15:10:08 OCA PAD AMENDMENT - PROJECT HEADER INFORMATION 05/26/94

Active

Project #: E-16-M53 Cost share #: Rev #: 3
Center # : 10/24-6-R7596-0A0 Center shr #: OCA file #:
Work type : RES
Contract#: NCA2-747 Mod #: ADMINISTRATIVE Document : OTH
Prime #: Contract entity: GTRC

Subprojects ? : N CFDA: 43.002
Main project #: PE #: N/A

Project unit: AERO ENGR Unit code: 02.010.110
Project director(s):
WU J C AERO ENGR (404)894-3028
WANG C M AERO ENGR (404)-

Sponsor/division names: NASA / AMES RESEARCH CTR, CA
Sponsor/division codes: 105 / 006

Award period: 921101 to 941231 (performance) 941231 (reports)

Sponsor amount	New this change	Total to date
Contract value	0.00	70,559.00
Funded	0.00	70,559.00
Cost sharing amount		0.00

Does subcontracting plan apply ?: N

Title: STUDIES OF HIGH LIFT AIRFOILS WITH IMPROVED STALL CHARACTERISTICS

PROJECT ADMINISTRATION DATA

OCA contact: Anita D. Rowland 894-4820

Sponsor technical contact

DR. KENNETH MCALISTER
(415)604-5892

FLUID MECHANICS DIVISION
MAIL STOP 215-1
NASA-AMES RESEARCH CENTER
MOFFETT FIELD, CA 94035-1000

Sponsor issuing office

MS. BARBARA HASTINGS, ADMIN. OFFICER
(415)604-5238

NASA-AMES UNIVERSITY CONSORTIUM
MAIL STOP 223-9
NASA-AMES RESEARCH CENTER
MOFFETT FIELD, CA 94035-1000

Security class (U,C,S,TS) : U ONR resident rep. is ACO (Y/N): N
Defense priority rating : N/A N/A supplemental sheet
Equipment title vests with: Sponsor GIT
"PURCHASE OF EQUIPMENT OR FACILITIES IS NOT AN ALLOWABLE COST."
Administrative comments -
SPONSOR'S 5/19/94 LETTER AUTHORIZES A NO-COST EXTENSION THRU **31 DEC 94

GEORGIA INSTITUTE OF TECHNOLOGY
OFFICE OF CONTRACT ADMINISTRATION

NOTICE OF PROJECT CLOSEOUT

Closeout Notice Date 01/27/95

Project No. E-16-M53 Center No. 10/24-6-R7596-0A0

Project Director WU J C School/Lab AERO ENGR

Sponsor NASA/AMES RESEARCH CTR, CA

Contract/Grant No. NCA2-747 Contract Entity GTRC

Prime Contract No.

Title STUDIES OF HIGH LIFT AIRFOILS WITH IMPROVED STALL CHARACTERISTICS

Effective Completion Date 941231 (Performance) 941231 (Reports)

Closeout Actions Required:	Y/N	Date Submitted
Final Invoice or Copy of Final Invoice	Y	
Final Report of Inventions and/or Subcontracts	Y	
Government Property Inventory & Related Certificate	N	
Classified Material Certificate	N	
Release and Assignment	N	
Other	N	

Comments

Subproject Under Main Project No.

Continues Project No.

Distribution Required:

Project Director	Y
Administrative Network Representative	Y
GTRI Accounting/Grants and Contracts	Y
Procurement/Supply Services	Y
Research Property Managment	Y
Research Security Services	N
Reports Coordinator (OCA)	Y
GTRC	Y
Project File	Y
Other	N
	N

NOTE: Final Patent Questionnaire sent to PDPI.

FINAL REPORT

STUDIES OF HIGH LIFT AIRFOILS WITH IMPROVED STALL CHARACTERISTICS

Clin M. Wang and James C. Wu
School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA

And

Kenneth W. McAlister and Chee Tung
US ARMY AeroFlightDynamics Directorate
NASA Ames Research Center
Moffett Field, CA

Funds for the support of this study have been allocated
by the NASA-Ames Research Center
Moffett Field, CA
Under
Interchange No. NCA 2-747

Dec., 1994

Abstracts

The present NASA-University consortium involves research work on development of an efficient and accurate numerical procedure to treat general unsteady viscous flows. Of particular interest in the present work are flows relevant to helicopter rotors. Special efforts are centered on new concepts that are applied to the helicopter's rotor blade configuration to improve the rotor aerodynamic characteristics. Among them, airfoil dynamic stall and the aerodynamic noise of vortex-airfoil interactions were extensively studied using the numerical procedure.

The procedure, based on the formulation of an integral representation of the velocity vector and the vorticity transport equation, is used to solve incompressible Navier-Stokes flows. A computer code ZETA, which has been developed previously at Georgia Tech, is used as the basis for the development of an extended computer code, ZETA II. The ZETA code has been demonstrated to be very efficient in the treatment of single solid airfoils undergoing arbitary motions. The extended ZETA II can treat the two-element airfoil, deformable airfoil, and vortex-airfoil interaction.

New concepts aimed at improving the helicopter maneuverability and susceptibility are studied. Concepts studied here are slatted airfoil and the deformable airfoil. Numerical results obtained by using the ZETA II code show that both slatted and deformable airfoils reduce the dynamic stall and that the deformable airfoil reduces the BVI (Blade-Vortex Interaction) noise.

TABLE OF CONTENTS

Introduction

The development of new generation helicopters to operate at much higher performance level than that in the past requires advanced design concepts and demonstrations. The new generation helicopters must perform well, particularly, in the areas of nap-of-the-earth, deep-penetration operations, and air-to-air combat. The technology base in order to meet the requirements of a highly maneuverable, agile, and survivable rotorcraft covers the fields of aerodynamics, acoustics, dynamics, and flight controls. Based on the currently available technology, the US Army is extending the technology base by a demonstration program, called the Helicopter Active Control Rotor (HACR) program. The objectives of the program include an increase in the capability of maneuverability/agility by 50% and a reduction in the acoustic detection range by 50%.

The most important element of the rotorcaft for providing the lift, control, and speed is the main rotor itself. The rotor is also a major source of acoustically detectable radiation. To achieve the capabilities of the new generation helicopters, the main R/D effort here will be placed on the demonstration of new concepts for rotor blade design and their effectiveness. In particular, the maneuvering capability can be improved by reducing or suppressing the vibratory loads on the rotor blades caused by flow separation and therefore the dynamic stall. Ways that suppress dynamic stall would have the effect of expanding the stall-limiting boundary of the rotor and thereby increases the available load factor in flight regimes. The rotor's vibratory loads are caused by the blade's angle of attack change, which is small on the advancing side and large on the retreating side. During a dash or a rapid maneuver, the rotor blades encounter stall on the retreating side, causing a rapid loss in lift and a sudden increase in both drag and pitching-moment.

The implusive noise generated by the rotor can be catagorized according to two main sources: the compressible effect near the blade tip and the BVI (Blade-Vortex Interaction) in descending or maneuvering flight. The BVI noise is most critical near the leading-edge of a blade when the vortex trajectory is nearly parallel to the blade span. The strong effect of the passing vortex is manifested by a large fluctuation of the surface pressure as the vortex passes closely to the leading-edge. This large fluctuation is the source of the BVI noise.

The requirement for improved maneuverability and reduced susceptibility clearly demands a substantial growth in technologies that address rotor aerodynamics. New control concepts, both passive and active which have the potential to meet the new requirements, must be considered with a more thorough physical understanding of flow phenomena as described earlier. Substantially improved prediction capabilities are key elements to provide the understanding the flow phenomena and to predict the aerodynamic loads. Various concepts for improving rotor aerodynamic characterisitics, including airfoils with a slat and slot, airfoils with flow energizers, and deformable airfoils, have been proposed by Army AFDD researchers.

The main objective of the present study is to demonstrate the effect of new concepts in reducing the dynamic stall and the BVI noise. In order to do this, efforts are placed on the development of a computer code that can efficiently predict flow phenomena and aerodynamic loads. These flow problems are directly related to the implementation of the new concepts. During the past, Georgia Tech researchers have developed a computer code ZETA that can predict complex flow phenomena of airfoils undergoing arbitary motion. It predicts airfoil load hysteresis loops of dynamic stall to a degree that matches the available experimental data within the experimental error band. The code is based on a vorticity-velocity formulation and an integral formulation of the velocity vector. The efficiency is attributible to the confinement of the computation to the vortical region only. The large potential region surrounding the vortical region does not enter into the computation procedure. For problems of practical interest, the size of the vortical region is much smaller than the size of the potential region.

Furthermore, the vortical region can be partitioned into attached and detached flow zones. In the attached flow zones, the flow is attached to soild surfaces and the boundary layer approximations are justified. Therefore the boundary layer equations are solved in these zones. The Navier-Stokes equations are solved in the detached flow zones, which consist of the separated and the wake flow zones. An iterative procedure is required only in the detached flow zones whereas a marching procedure is used in the attached flow zones. The marching procedure takes very little time compared to the time consumed for an iterative procedure. The high efficiency of the numerical zonal method permits an effective way for conducting a parametric study of the new concepts to be implemented on helicopter rotors.

The numerical procedure of the ZETA code is extended in the present effort for treating problems of a broader scope. A numerical conformal procedure is established first such that two-element airfoil geometries and deformable airfoils can be handled. The conformal mapping technique is an important element of the present numerical procedure because the conformal mapping leads to the simplest possible form of the governing equations to be solved and the integral formulation derived for the velocity can be directly used. The ZETA code was developed for solid single airfoil and uses conformal mappings techniques based on an analytical method. In the extended code ZETA II, conformal mapping is performed numerically such that it can treat two-element and deformable airfoils. The extension of the ZETA to ZETA II also includes numerical procedures that are made suitable for two-element and deformable airfoils. For BVI problems, a passing vortex is modeled by a single point vortex that travels closely to an airfoil. The high efficiency of the ZETA code is essentially retained in the extended ZETA II code.

Various new concepts for rotor blades are demonstrated by using the newly developed code. In the present study, the slatted airfoil and the deformable airfoil are extensively investigated in order to demonstrate an improvement on dynamic stall and BVI noise. Publications involving the present effort are included in Appendix A. A manual and the program listing of the ZETA II code are included as Appendix B.

Mathematical Formulations

Through the introduction of vorticity, a general unsteady viscous flow problem can be partitioned into its kinematic and kinetic aspects. The kinematic aspect describes the velocity field corresponding to the vorticity field at any instant of time, subjected to the velocity boundary condition. The kinetic aspect describes the transport of the vorticity field through convection and diffusion. The vorticity transport equation is elliptic in space and therefore the vorticity on the flow boundaries is required. The values of vorticity on solid boundaries, however, are not given directly from the physics of a flow problem. An accurate and unique determination of the boundary vorticity values is essential. In the present approach, the boundary vorticity values are determined from an integral constraint derived from the velocity integral representation. The theoretical foundation of the boundary vorticity evaluation was established by Wu [1].

Airfoils in the physical plane (z-plane) are conformally transformed into simple geometries in the computational plane (ζ-plane). By using conformal mapping, the integral representation of the velocity vector, derived in the physical plane, can be directly used in the computational plane. The vorticity transport equation is also in its simplest possible form in the computational plane. Also, the circulation of a passing vortex is not changed by the transformation between the two planes. A fast Fourier transform can be applied to the computational procedure, resulting in the present efficient computer code ZETA II.

Conformal Mapping of Two-Element Airfoils

The conformal mapping technique used here was introduced by Ives[2] and modified by Hall and Suddhoo[3]. In particular, a slat-airfoil geometry is transformed from the physical plane to two concentric circles in the transformed plane. The following mapping steps are involved in this transform:

1. A Karman-Trefftz transformation which maps the slat onto a near circle while the main airfoil deforms slightly.

2. Another Karman-Trefftz transformation which maps the main airfoil configuration onto a near circle while the near circle corresponding to the slat deforms slightly.

3. A Theodosen numerical transformation which maps the near circle corresponding to the main airfoil onto a unit circle. The near circle corresponding to the slat deforms slightly again. The near circle is located outside the unit circle.

4. A bilinear transformation which maps the near circle from outside the unit circle to a near circle inside the unit circle. The external flow problem is now converted into an

internal flow problem. The farfield boundary of the external flow problem is mapped onto a singular point located on the real axis in the annular region.

5. A Garrick-Theodosen numerical transformation which maps the near circle onto a circle. The singular point will in general move off from the real axis.

6. A rotation which brings the singular point back to the real axis in the transformed plane.

Notice that the transformation steps can be used to transform any two-element airfoil geometry, e.g., an airfoil-flap configuration. For problems with a point vortex moving in the fluid, the location of the vortex at any instant of time can be easily correlated between the physical and the computational plane. The transformation procedure contains a simpler procedure that transforms a single airfoil onto a unit circle. This procedure allows an airfoil to perform different types of deformation and the shape be transformed conformally. In the simpler procedure, only Steps 2, 3 and 4 are needed and the singular point is located at the origin of the computational plane.

Kinematics

The integral representation of the velocity vector in the ζ-plane (in cylindrical coordinates ρ, ϕ), with a passing point vortex of circulation $\vec{\Gamma}_P$ at a location $\vec{\rho}_P$, is:

$$\begin{aligned} \vec{v}(\vec{\rho}) \;=\; & -\int_R \vec{\omega}_o H_o^2 \times \vec{Q} dR_o \\ & +\int_B [(\vec{v}_o \cdot \vec{n}_o) - (\vec{v}_o \times \vec{n}_o)\times]\vec{Q} dB_o \\ & +\frac{1}{2\pi}\frac{\vec{\Gamma}_P \times (\vec{\rho} - \vec{\rho}_P)}{|\vec{\rho} - \vec{\rho}_P|^2} \end{aligned} \tag{1}$$

where $\vec{\omega}$ is the vorticity vector; H is the scale factor of the transformation, $H = |dz/d\zeta|$; B is the boundary of the fluid region R; $\vec{n}$ is a unit normal vector on B directed outward from R; the subscript "o" indicates that a variable or an integration is in the $\vec{\rho}_o$ space; $\vec{Q}$ is the gradient of the fundamental solution of elliptic equations,

$$\vec{Q} = \frac{\vec{\rho}_o - \vec{\rho}}{2\pi|\vec{\rho}_o - \vec{\rho}|^2}$$

The boundary integral in Equation (1) represents the velocity boundary contributions. It includes the solid body motions and the free stream velocity if the reference frame is moving with the solid bodies at a velocity, $-\vec{v}_\infty$, where $\vec{v}_\infty$ is the free stream velocity. It may also include the surface velocity introduced by the body deformation if the body undergoes deformable motion.

In the transformed plane, the singular point introduced by the bilinear transformation represents the farfield boundary infinitely far from the solid bodies. The boundary B in Equation (1) therefore must include an infinitesmal circle surrounding the singular point. The boundary integral around this small circle can be analytically evaluated since the velocity on this circle is known through the conformal transformation. After the anaytical integration on this circle, the singular point is therefore mathematically excluded from the computational plane. In actual computations, a small but finite region around the singular point is removed. In the physical plane, the removal of this singular region represents the computation region in a finite region surrounding the solids. The excluded region in general contains vorticity. For example, for a steady flow around an airfoil at a small angle of attack, the excluded region contains the starting vortices that are shed from the vicinity of the airfoil. For a general unsteady flow, the region contains no vorticity at the beginning but may contain vorticity later on as vorticity shed by the airfoil reaches this region. The excluded region therefore may possess a circulation. In the transformed plane, the singular region serves as a vortex sink which draws the vorticity shed earlier from the vicinity of the solids. The circulation of the excluded region contributes to the velocity field in the transformed plane in accordance with the Biot-Savart law. Taking the above into consideration, one has from Equation (1),

$$\begin{aligned}\vec{v} \;=\; & -\int_R \vec{\omega}_o H_o^2 \times \vec{Q} dR_o + \vec{v}_S + \frac{\Gamma_S}{2\pi}\vec{v}_\Gamma \\ & + \int_{S_1+S_2} [(\vec{v}_o \cdot \vec{n}_o) - (\vec{v}_o \times \vec{n}_o)\times]\vec{Q} dB_o \\ & + \frac{\Gamma_P}{2\pi}\vec{v}_P\end{aligned} \tag{2}$$

where the integrals over S_1 and S_2 represent the velocity contributions from the solid motion of the slat and the main airfoil respectively, for the case of a slatted airfoil; Γ_S is the circulation of the excluded region; $\vec{v}_\Gamma$ represents the contribution from Γ_S per unit circulation; and $\vec{v}_S$ is the contribution from the free stream that has been integrated analytically. Consider a bilinear transformation

$$z = \frac{1 - \bar{a}\zeta}{\zeta - \bar{a}} \tag{3}$$

which transforms the infinity in the z-plane to the point located at $\bar{a}$ on the real axis of the ζ-plane, one has

$$\begin{aligned}\vec{v}_S = v_{S_\rho}\vec{e}_\rho + v_{S_\phi}\vec{e}_\phi = \frac{v_\infty(\bar{a}-1)}{(\rho^2+\bar{a}^2-2\bar{a}\rho cos\phi)^2} \times \\ \{(cos\alpha_\infty[(\bar{a}^2+\rho^2)cos\phi - 2\bar{a}\rho] + sin\alpha_\infty(\bar{a}^2-\rho^2)sin\phi)\vec{e}_\rho \\ +(sin\alpha_\infty[(\bar{a}^2+\rho^2)cos\phi - 2\bar{a}\rho] - cos\alpha_\infty(\bar{a}^2-\rho^2)sin\phi)\vec{e}_\phi\}\end{aligned} \tag{4}$$

where $\vec{e}_\rho$ and $\vec{e}_\phi$ denote the unit directional vectors in the ζ-plane; α_∞ is the angle of attack. In the present problem, the contribution from the singular point retains the same form of Equation (4) but is modified due to a different bilinear transformation used in Step 4 of the mapping procedure. The modifications also include the effects of the scale factors that have been introduced through the mapping steps other than the bilinear transformation.

The velocity vector $\vec{v}_\Gamma$ in Equation (2) is

$$\begin{aligned} \vec{v}_\Gamma &= v_{\Gamma_\rho}\vec{e}_\rho + v_{\Gamma_\phi}\vec{e}_\phi \\ &= \frac{-\bar{a}sin\phi}{\rho^2+\bar{a}^2-2\bar{a}\rho cos\phi}\vec{e}_\rho + \frac{\rho-\bar{a}cos\phi}{\rho^2+\bar{a}^2-2\bar{a}\rho cos\phi}\vec{e}_\phi \end{aligned} \tag{5}$$

Fourier Series Expansion

Let the two components of the velocity vector be v_ρ and v_ϕ. From Equation (2), one has

$$\begin{aligned} v_\rho &= \frac{1}{2\pi}\int_R \frac{\omega_o H_o^2 \rho_o sin(\phi_o-\phi)}{\rho_o^2+\rho^2-2\rho_o\rho cos(\phi_o-\phi)}\rho_o d\rho_o d\phi_o \\ &+\frac{1}{2\pi}\int_{S_1+S_2}\frac{v_{\rho_o}[\rho_o cos(\phi_o-\phi)-\rho]}{\rho_o^2+\rho^2-2\rho_o\rho cos(\phi_o-\phi)}\rho_o d\phi_o \\ &-\frac{1}{2\pi}\int_{S_1+S_2}\frac{v_{\phi_o}\rho_o sin(\phi_o-\phi)}{\rho_o^2+\rho^2-2\rho_o\rho cos(\phi_o-\phi)}\rho_o d\phi_o \\ &+v_{\rho_s}+\frac{\Gamma_S}{2\pi}v_{\Gamma_\rho}+\frac{\Gamma_P}{2\pi}v_{P_\rho} \end{aligned} \tag{6}$$

$$\begin{aligned} v_\phi &= -\frac{1}{2\pi}\int_R \frac{\omega_o H_o^2[\rho_o cos(\phi_o-\phi)-\rho]}{\rho_o^2+\rho^2-2\rho_o\rho cos(\phi_o-\phi)}\rho_o d\rho_o d\phi_o \\ &+\frac{1}{2\pi}\int_{S_1+S_2}\frac{v_{\rho_o}\rho_o sin(\phi_o-\phi)}{\rho_o^2+\rho^2-2\rho_o\rho cos(\phi_o-\phi)}\rho_o d\phi_o \\ &+\frac{1}{2\pi}\int_{S_1+S_2}\frac{v_{\phi_o}[\rho_o cos(\phi_o-\phi)-\rho]}{\rho_o^2+\rho^2-2\rho_o\rho cos(\phi_o-\phi)}\rho_o d\phi_o \\ &+v_{\phi_s}+\frac{\Gamma_S}{2\pi}v_{\Gamma_\phi}+\frac{\Gamma_P}{2\pi}v_{P_\phi} \end{aligned} \tag{7}$$

If the velocity components and the vorticity are expanded in Fourier series as follows:

$$v_\rho = \frac{a_0(\rho)}{2} + \sum_1^N [a_n(\rho)cosn\phi + b_n(\rho)sinn\phi]$$

$$
\begin{aligned}
v_\phi &= \frac{c_0(\rho)}{2} + \sum_1^N [c_n(\rho) cosn\phi + d_n(\rho) sinn\phi] \\
\omega H^2 &= \frac{\alpha_0(\rho)}{2} + \sum_1^N [\alpha_n(\rho) cosn\phi + \beta_n(\rho) sinn\phi] \\
v_{S_\rho} &= \frac{x_0(\rho)}{2} + \sum_1^N [x_n(\rho) cosn\phi + y_n(\rho) sinn\phi] \\
v_{S_\phi} &= \frac{s_0(\rho)}{2} + \sum_1^N [s_n(\rho) cosn\phi + t_n(\rho) sinn\phi] \\
v_{\Gamma_\rho} &= \frac{e_0(\rho)}{2} + \sum_1^N [e_n(\rho) cosn\phi + f_n(\rho) sinn\phi] \\
v_{\Gamma_\phi} &= \frac{g_0(\rho)}{2} + \sum_1^N [g_n(\rho) cosn\phi + h_n(\rho) sinn\phi] \\
v_{P_\rho} &= \frac{p_0(\rho)}{2} + \sum_1^N [p_n(\rho) cosn\phi + q_n(\rho) sinn\phi] \\
v_{P_\phi} &= \frac{i_0(\rho)}{2} + \sum_1^N [i_n(\rho) cosn\phi + j_n(\rho) sinn\phi]
\end{aligned} \tag{8}
$$

Substituting Equation (8) into Equations (6) and (7) and then performing analytical integration on ϕ-direction, one obtains the velocity Fourier coefficients in the following,

$$
a_0(\rho) = a_0(\rho_S) \cdot \frac{\rho_S}{\rho} + x_0(\rho) + \frac{\Gamma_S}{2\pi} e_0(\rho) + \frac{\Gamma_P}{2\pi} p_0(\rho)
$$

$$
\begin{aligned}
a_n(\rho) &= \frac{1}{2}\int_{\rho_S}^{\rho} \beta_n(\rho_o) \cdot (\frac{\rho_o}{\rho})^{n+1} d\rho_o + \frac{1}{2}\int_{\rho}^{1} \beta_n(\rho_o) \cdot (\frac{\rho}{\rho_o})^{n-1} d\rho_o \\
&+ \frac{1}{2} a_n(1) \cdot \rho^{n-1} + \frac{1}{2} a_n(\rho_S) \cdot (\frac{\rho_S}{\rho})^{n+1} - \frac{1}{2} d_n(1) \cdot \rho^{n-1} \\
&+ \frac{1}{2} d_n(\rho_S) \cdot (\frac{\rho_S}{\rho})^{n+1} + x_n(\rho) + \frac{\Gamma_S}{2\pi} e_n(\rho) + \frac{\Gamma_P}{2\pi} p_n(\rho) \\
b_n(\rho) &= -\frac{1}{2}\int_{\rho_S}^{\rho} \alpha_n(\rho_o) \cdot (\frac{\rho_o}{\rho})^{n+1} d\rho_o - \frac{1}{2}\int_{\rho}^{1} \alpha_n(\rho_o) \cdot (\frac{\rho}{\rho_o})^{n-1} d\rho_o \\
&+ \frac{1}{2} b_n(1) \cdot \rho^{n-1} + \frac{1}{2} b_n(\rho_S) \cdot (\frac{\rho_S}{\rho})^{n+1} + \frac{1}{2} c_n(1) \cdot \rho^{n-1} \\
&- \frac{1}{2} c_n(\rho_S) \cdot (\frac{\rho_S}{\rho})^{n+1} + y_n(\rho) + \frac{\Gamma_S}{2\pi} f_n(\rho) + \frac{\Gamma_P}{2\pi} q_n(\rho) \qquad (9) \\
c_0(\rho) &= \int_{\rho_S}^{\rho} \alpha_0(\rho_o) \cdot \frac{\rho_o}{\rho} d\rho_o + c_0(\rho_S) \cdot \frac{\rho_S}{\rho} + s_0(\rho) + \frac{\Gamma_S}{2\pi} g_0(\rho) + \frac{\Gamma_P}{2\pi} i_0(\rho) \\
c_n(\rho) &= \frac{1}{2}\int_{\rho_S}^{\rho} \alpha_n(\rho_o) \cdot (\frac{\rho_o}{\rho})^{n+1} d\rho_o - \frac{1}{2}\int_{\rho}^{1} \alpha_n(\rho_o) \cdot (\frac{\rho}{\rho_o})^{n-1} d\rho_o
\end{aligned}
$$

$$
\begin{aligned}
& +\frac{1}{2}b_n(1)\cdot\rho^{n-1}-\frac{1}{2}b_n(\rho_S)\cdot(\frac{\rho_S}{\rho})^{n+1}+\frac{1}{2}c_n(1)\cdot\rho^{n-1} \\
& +\frac{1}{2}c_n(\rho_S)\cdot(\frac{\rho_S}{\rho})^{n+1}+s_n(\rho)+\frac{\Gamma_S}{2\pi}g_n(\rho)+\frac{\Gamma_P}{2\pi}i_n(\rho) \\
d_n(\rho) & =\frac{1}{2}\int_{\rho_S}^{\rho}\beta_n(\rho_o)\cdot(\frac{\rho_o}{\rho})^{n+1}d\rho_o-\frac{1}{2}\int_{\rho}^{1}\beta_n(\rho_o)\cdot(\frac{\rho}{\rho_o})^{n-1}d\rho_o \\
& -\frac{1}{2}a_n(1)\cdot\rho^{n-1}+\frac{1}{2}a_n(\rho_S)\cdot(\frac{\rho_S}{\rho})^{n+1}+\frac{1}{2}d_n(1)\cdot\rho^{n-1} \\
& +\frac{1}{2}d_n(\rho_S)\cdot(\frac{\rho_S}{\rho})^{n+1}+t_n(\rho)+\frac{\Gamma_S}{2\pi}h_n(\rho)+\frac{\Gamma_P}{2\pi}j_n(\rho)
\end{aligned}
$$

In the above equations, $1 \leq n \leq N$ and $a_0(\rho_S)$, $a_n(\rho_S)$, $b_n(\rho_S)$ ... etc., represent the Fourier coefficients of the velocity boundary conditions at $\rho = \rho_S$. The small circle of radius ρ_S corresponds to the slat in the physical plane. The velocity Fourier coefficients, $a_0(1)$, $a_n(1)$, $b_n(1)$ ... etc., are the velocity boundary conditions at $\rho = 1$ which denotes the main airfoil. For a single airfoil case, the above equations are simplified by dropping the terms containing ρ_S.

The usage of the Fourier series offers important advantages in the numerical process. It not only increases the solution accuracy but also reduces the computation work drastically. Each velocity Fourier coefficient relates to the corresponding vorticity Fourier coefficient by a simple one-dimensional integral. Also, the integral relations, Equation (9), give explicitly a series of constraint equations on the vorticity Fourier coefficients which allows the explicit evaluation of the vorticity values on the solid surfaces.

Vorticity Integral Constraints

The vorticity values on the solid surfaces are determined by satisfying the velocity boundary conditions. If one applies Equation (2) on the solid surfaces S_1 and S_2, constraint equations for the vorticity field result. This is because the velocity on the surface is given and the velocities induced by Γ_S and Γ_P are known. The vorticity field is therefore restricted by the domain integral. Using the Fourier series, the constraint equations for the vorticity Fourier coefficients when at $\rho = \rho_S$ are found from Equation (9) to be,

$$
\begin{aligned}
\int_{\rho_S}^{1}\alpha_n(\rho_o)\cdot(\frac{\rho_S}{\rho_o})^{n-1}d\rho_o &= b_n(1)\cdot\rho_S^{n-1}-b_n(\rho_S)+c_n(1)\cdot\rho_S^{n-1}-c_n(\rho_S) \\
& \quad +2s_n(\rho_S)+\frac{\Gamma_S}{\pi}g_n(\rho_S)+\frac{\Gamma_P}{\pi}q_n(\rho_S) \\
\int_{\rho_S}^{1}\beta_n(\rho_o)\cdot(\frac{\rho_S}{\rho_o})^{n-1}d\rho_o &= -a_n(1)\cdot\rho_S^{n-1}+a_n(\rho_S)+d_n(1)\cdot\rho_S^{n-1}-d_n(\rho_S) \\
& \quad +2t_n(\rho_S)+\frac{\Gamma_S}{\pi}h_n(\rho_S)+\frac{\Gamma_P}{\pi}j_n(\rho_S)
\end{aligned}
\qquad (10)
$$

At $\rho = 1$, one has

$$\begin{aligned}
\int_{\rho_S}^{1} \alpha_n(\rho_o)\cdot(\rho_o)^{n+1}d\rho_o &= c_n(1) - c_n(\rho_S)\cdot\rho_S^{n+1} - b_n(1) + b_n(\rho_S)\cdot\rho_S^{n+1} \\
&\quad -2s_n(1) - \frac{\Gamma_S}{\pi}g_n(1) - \frac{\Gamma_P}{\pi}q_n(1) \\
\int_{\rho_S}^{1} \beta_n(\rho_o)\cdot(\rho_o)^{n+1}d\rho_o &= d_n(1) - d_n(\rho_S)\cdot\rho_S^{n+1} + a_n(1) - a_n(\rho_S)\cdot\rho_S^{n+1} \\
&\quad -2t_n(1) - \frac{\Gamma_S}{\pi}h_n(1) - \frac{\Gamma_P}{\pi}j_n(1)
\end{aligned} \tag{11}$$

Once the values for α_n and β_n are known in the interior of the flow domain, Equations (10) and (11) give a unique determination of α_n and β_n, where $1 \leq n \leq N$, at both $\rho = \rho_S$ and $\rho = 1$. Notice that the above integral constraints are obtained by either satisfying the boundary condition on v_ϕ, the tangential velocity component, or v_ρ, the normal velocity component, on the solid surfaces.

To determine α_0 at $\rho = \rho_S$ and at $\rho = 1$, however, one needs additional information. If the individual circulations around the two closed paths which enclose the slat and the main airfoil in the ζ-plane are known, one has

$$\begin{aligned}
\int_{R_1} \omega H^2 dR = \Gamma_1 \\
\int_{R_2} \omega H^2 dR = \Gamma_2
\end{aligned} \tag{12}$$

where R_1 and R_2 denote the flow domain enclosed by the two paths around the slat and the main airfoil respectively. The circulations Γ_1 and Γ_2 are known by tracking the vorticity flowing out of R_1 and R_2 at every time step. The principle of conservation of total vorticity gives

$$\Gamma_1 + \Gamma_2 = -\Gamma_3 \tag{13}$$

where Γ_3 includes the integrated value of ωH^2 in the remaining computational domain and Γ_S. Using Equation (8) on the Fourier series expansion of ωH^2, one gets the constraints for α_0 from Equation (12)

$$\begin{aligned}
\int_{\rho_S}^{\rho_1} \alpha_0(\rho_o)\cdot\rho_o d\rho_o = \frac{\Gamma_1}{\pi} \\
\int_{\rho_2}^{1} \alpha_0(\rho_o)\cdot\rho_o d\rho_o = \frac{\Gamma_2}{\pi}
\end{aligned} \tag{14}$$

where ρ_1 and ρ_2 represent the radii of the two individual closed paths in the ζ-plane.

Kinetics

The kinetic aspect of the flow problem is described by the vorticity transport equation. In the computational plane, the equation is

$$H^2 \frac{\partial \vec{\omega}}{\partial t} = \nabla \times (\vec{v} - \vec{v}_G) \times \vec{\omega} + \nabla \cdot (\nu \nabla \vec{\omega}) \tag{15}$$

where t denotes the time and ν represents the kinematic viscosity of the fluid. In turbulent flow calculations, ν includes an eddy viscosity which is computed here by the Baldwin and Lomax algebraic model [4]. The velocity $\vec{v}_G$ represents the grid velocity observed in the computational plane during the airfoil deformation. In the attached flow zones, the last term of Equation (15) is simplified to $\partial/\partial n(\nu \partial \omega/\partial n)$, where n is the normal coordiante perpendicular to the streamwise direction. The vorticity transport equation becomes a parabolic type in the attached flow zones. For problems containing a passing point vortex, it travels in the flowfield with a local velocity that is induced by all the distributed vorticity existing in the flowfield and also by the circulation Γ_S. The circulation Γ_P carried by the point vortex is unchanged during its movement in the fluid.

One of the advantages of using the vorticity transport equation is that the pressure is eliminated from the mathematical formulation. The pressure on solid surfaces, wherever needed, is usually obtained from the normal gradient of the computed vorticity on the surface[5]. With this approach, an accurate determination of the normal gradient of the vorticity field is a difficult task when both the magnitude and the gradient of the boundary vorticity values are large at high Reynolds numbers. In particular, for turbulent flow computation, the modelling of flow turbulence usually requires the assumptions of different sub-layers near the solid surfaces. Substantial uncertainties and inaccuracies in the vorticity normal grandient are implicit in such models. To eliminate these difficulties, the present authors reformulated the pressure as an integral representation by using a principal solution of the Poisson equation[6]. It is demonstrated in Reference 6 that this formulation yields accurate pressure solutions. In the present study, this formulation is extended for multi-element airfoils.

Numerical Formulations

In the present computation, the airfoil is first transformed to a simple geometry in the computational plane. The computations are performed in the computational plane. The computational grid is generated by the numerical conformal mapping with proper control on both radial and circumferential directions in the computational plane. If the airfoil undergoes time-dependent deformation, the conformal mapping is performed at every time step. The following steps constitute a complete computation cycle which advances the velocity and the vorticity solutions from an old time level to a new time level:

1. Perform the numerical conformal mapping when the airfoil deforms. The grid velocity $\vec{v}_G$ is numerically evaluated in the physical plane and then transformed to the computational plane.

2. Compute the convection velocity $\vec{v}_P$ of the point vortex in the computational plane. This is the velocity that is induced by all the distributed vorticity existing in the flowfield and also by Γ_S. The distance traveled by the point vortex during a time step Δt is $\vec{v}_P \Delta t / H^2$. This determines the position of the point vortex at the new time level.

3. Compute the distributed vorticity, excluding the surface vorticity, for the new time level by solving the vorticity transport equation, (15).

4. Determine the surface vorticity where the new distributed vorticity is obtained by Step 3 and the new location of the point vortex is known from Step 2.

5. Compute the velocity in the vortical region for the new time level by using Equation (9).

In Step 3 the vorticity values in the detached flow zones are obtained by solving a fully implicit finite-difference equation discretized from the vorticity transport equation. The time derivative is approximated by an one-step forward difference scheme and the diffusion terms are approximated by central difference. The convection terms are approximated by a second upwind difference scheme [5]. In the present procedure, a line-relaxation technique is used to solve the simultaneous implicit finite-difference equations. The line-relaxation involves an inversion of a tri-diagonal matrix where the unknown vorticity values on the same radial line are to be solved. The vorticity values on the adjacent radial lines are treated as known values. These values are delayed by one iteration count in the relaxation procedure. During the iteration process, an under-relaxation factor is found necessary.

The vorticity values in the attached flow zones are computed by a marching procedure. The detached and the attached flow zones are divided by demarkation lines. The vorticity values on the demarkation lines are the initial condition for the vorticity in the attached zones and are part of the boundary conditions for the detached zones.

In Step 4 the boundary vorticity values adjacent to solid surfaces are obtained through the integral constraint of the vorticity Fourier coefficients. The converged interior vorticity values obtained in Step 3 are first transformed to the Fourier coefficients, in accordance with the ωH^2 expression in Equation (8), on the circumferential direction at each constant radial station. For example,

$$\alpha_n = \frac{1}{\pi} \int_0^{2\pi} \omega H^2 cos(n\phi) d\phi \tag{16}$$

The integral in Equation (16) is replaced by a numerical quadrature. Each vorticity Fourier coefficient, except the constant coefficient, is subjected to the two associated integral constraint equations, Equations (10) and (11). This allows the two values of the Fourier coefficient at $\rho = \rho_S$ and at $\rho = 1$ to be determined uniquely. The constant Fourier coefficient α_0, as stated earlier, is determined by the conservation of vorticity. That is, once the quantities of Γ_1 and Γ_2 are known in Equation (12), the boundary values of this Fourier coefficient are determined by the two separated expressions in Equation (12). In the present unsteady approach, the quantities Γ_1 and Γ_2 are tracked at each time level by recording the amount of vorticity flowing out of the regions R_1 and R_2, respectively. After knowing the Fourier coefficients at the boundaries, the vorticity values at the boundaries can be easily determined from the ωH^2 expression in Equation (8).

Steps 3 and 4 are repeated several times before the numerical procedure progresses to Step 5. The updated boundary vorticity values may not equal the boundary values that Step 3 has previously used to compute the interior vorticity values. Once the maximum difference between the updated boundary vorticity values and the previous boundary values is less than a prescribed criterion, the computation then advances to Step 5. It is found that to make the process convergent, an under-relaxation on the boundary vorticity values is necessary. In Step 5 the interior velocity Fourier coefficients are calculated by numerical quadrature of the vorticity Fourier coefficients, Equation (9). That is, only one-dimensional quadrature is involved for each of the velocity Fourier coefficients. The velocity values are then obtained by the v_ρ and v_ϕ expressions in Equation (8).

The numerical conformal mapping procedures are coded in a separate computation module. This module is called only once when the airfoil does not deform, but is called every time step if the airfoil undergoes time-dependent deformation. The main module of the ZETA II code contains the numerical procedures as described by Steps 1 to 5. The surface pressure is computed by another module that reads in the velocity and vorticity obtained by the main module. The module that computes surface pressure also computes skin friction and aerodynamic loads. A detailed description of these modules is contained in Appendix B of the present report.

The ZETA II code is used to demonstrate new concepts that are useful to improve the rotor aerodynamic characterisitics. The present effort is focusd on problems of dynamic stall and BVI noise. The effects of new concepts are demonstrated by comparing the numerical results of the basic single airfoil with the results of the slatted airfoil or the defomable airfoil.

Dynamic Stall

There are several concepts that have been pursued by Army researchers at AFDD to descrease and even to eliminate the undesirable dynamic stall. These concepts can be categorized by active and passive devices. The passive devices include airfoils with an added slat or slot geometry, an airfoil with blowing/suction on the leeward side. The active device is mainly the airfoil with different types of deformation. In the present study, the slatted airfoil and several types of deformable airfoil are investigated by the numerical procedure. A Boeing VR-7 airfoil is used for the demonstration.

Slatted Airfoils

Flow solutions around the basic single VR-7 airfoil and the slatted VR-7 airfoil were obtained by the ZETA II code. The Reynolds number based on the airfoil chord is 200,000. A steady flow is fully established around the airfoil at a small angle of attack before the airfoil undergoes oscillating motion. The reduced frequency, normalized by the airfoil half chord and the oscillating frequecy, is 0.1 in the results reported here.

The grids used to predict the load characteristics are 81X51 and 121X61 in size for the basic and the slatted airfoils, respectively. Fig. 1 shows sample grids around the single and the slatted airfoil. A close-up view of the grids and the slat geometry is also shown in the figure. The pitching axis for both the basic and the slatted VR-7 airfoils is on the quarter chord of the main airfoil and the airfoil oscillates sinusoidally from 5 to 25°. The experimental hysteresis loops for the lift, drag and pitching moment coefficients are shown in Fig. 2, measured by AFDD researchers in their water tunnel. The dynamic stall of the basic VR-7 airfoil (solid line) is quite evident. The lift coefficient curve increases until reaching an angle of attack around 19° and maintains a constant value for about one degree. Then the lift coefficient increases again due to the formation of a leading-edge stall vortex. The airfoil experiences a large nose-down pitching moment as the vortex moves over the upper surface. When the vortex leaves the airfoil, the lift drops drastically. As for the slatted VR-7 airfoil (dash line), there is no stall vortex formed around the leading edge. The pitching moment shows no sign of a rapid change at higher angles of attack and no abrupt drop in the lift occurs.

The numerical solution for both airfoils starts at $\alpha = 5^o$ as an impulsive motion. When the solution converges, the oscillating motion of the airfoil is allowed to begin. The predicted lift, drag and moment hysteresis for the basic airfoil are shown in Fig. 3. The shape of each loop compares well with the test data, however, the predicted lift loop is consistently lower than the test results. The computed streamline patterns at selected angles of attack are shown in Fig. 4. The lift and pitching moment loops are shown again in this figure to demonstrate the flow features in connection with the dynamic stall. The formation of a leading-edge stall vortex and its movement from $\alpha = 22.93^o$ to 24.34^o correspond to the rise in lift and the large change in pitching moment. When the stall vortex leaves the airfoil, the lift drops sharply. The flow remains separated during the down-stroke portion of the cycle until $\alpha \approx 12^o$, but the lift and the pitchng moment are maintained approximately at the same level thoughout the down-stroke period. The instantaneous streamlines and the surface pressure for the slatted airfoil are shown in Fig. 5, with the dotted line in the C_P plot representing the slat and the solid line for the main airfoil. The shape of the hysteresis loops are quite similar to those of the test results in Fig. 2. The slatted airfoil does not have the formation of the leading-edge vortex. The flow around the leading edge of the main airfoil remains attached during the complete cycle. The flow starts to separate from the airfoil's trailing-edge at about $\alpha = 15^o$ and the separated flow never propagates to the leading edge.

Deformable Airfoils

There are three types of the airfoil deformation considered in the present study: (1). airfoil nose droop down; (2). airfoil camber change; and (3). airfoil thickness change. A representative of each type of deformed airfoil shape is shown in Fig. 6. The steady flow around the basic airfoil was first obtained at $\alpha = 5^o$ as an impulsive motion. The airfoil then starts to oscillate between 5^o and 25^o. During the oscillation motion, the airfoil deforms sinusoidally in proportion to the angle of attack. That is, the airfoil has no deformation at the beginning of the oscillation and has maximum deformation at the maximum angle of attack. The computed cases are for flows at a reduced frequency of 0.15 and Reynolds number of 1 million. The flow is assumed to be fully turbulent.

Fig. 7 shows the surface pressure and instantaneous streamlines at selected angles of attack during the oscillation cycle of the basic airfoil. The separation streamline starts from the airfoil's trailing-edge and moves upstream as the angle of attack increases. At 25°, the separation line almost reaches the airfoil's leading-edge. A leading-edge vortex is present at this angle of attack, but the lift drop of the airfoil begins at a much earlier angle, approximately at 21.5°. The load hysteresis loops are shown in Fig. 8. The initial lift drop is caused from the shedding of the separation bubble generated from the airfoil's trailing-edge. A sharp lift increase follows the initial lift drop is caused by the formation and growing of the leading-edge vortex. It then merges with the trailing-edge separation bubble and causes the abrupt second phase lift drop, or the lift stall. A large nose-down pitching moment is experienced by the airfoil near the maximum angle of attack. The difference between the loads hysteresis loops shown in Fig. 8 and those in Fig. 3 is mainly due to the difference in the reduced frequency.

With the airfoil nose drooping down, the instantaneous streamlines and surface pressure are shown in Fig. 9. In this figure, the front 37.5% of the airfoil rotates down as the angle of attack increases. The maximum deflection angle is 10°. No leading-edge vortex is present for the case shown here. This case is represented by (.375, 10.) where the first number denotes the portion of the airfoil that droops down and the second number denotes the maximum deflection angle. In the present study, three cases, (.25, 10.), (.375, 10.) and (.50, 10.), were computed. All three cases show that the leading-edge vortex is suppressed and therefore no deep lift stall is observed. Fig. 10 shows the load hysteresis loops. It is shown from the three cases computed that the (.375, 10.) case has the best dynamic characteristics.

The next deformation studied was for an airfoil that changes camber as the angle of attack changes. The airfoil attains maximum camber at the maximum angle of attack during the oscillation cycle. The maximum camber studied here is 5% of the airfoil chord. Three cases of different camber locations were computed: 40%, 50% and 60% of the chord. The instantaneous streamlines and the surface pressure are shown in Fig. 11 for the case where the camber is located at 50% chord location. The flow is similar to that of the previous nose droop-down case but the surface pressure is smoother for the cambered case. The load hysteresis loops for the three cambered cases are shown in Fig. 12. It can be seen that the 40% and the 50% cases show similar dynamic characteristics, but the 50% case generates higher lift. The lift stalls for the 60% case even though it is much milder than that for the basic airfoil. The pitching moment shows a behavior that is similar to the lift. The instantaneous streamlines for the 60% case, not shown here, indicate the separation streamline has reached near to the airfoil's leading-edge. The 50% case is the best among the cambered cases studied here.

The third airfoil deformation involves airfoil thickness change. The airfoil thickens in proportion to its original thickness distribution. The airfoil thickens to its maximum value at the maximum angle of attack during the airfoil oscillation. Three cases computed here are for increases in airfoil thickness by 25%, 50% and 75% of its original thickness. Fig. 13 shows the instantaneous streamlines and the surface pressure for the 50% thickened case. The airfoil thickening also suppresses the formation of the leading-edge vortex. The load hysteresis loops for the three computed cases are shown in Fig. 14. It shows the 25% case has a very mild lift stall and the other two cases are even milder. There is no moment stall for all the three cases. The 50% case has the best dynamic characteristics among the three cases studied.

Fig. 15 shows compares the three types of airfoil deformation and shows their affect on the dynamic stall. Also shown in the figure are the loads of the basic airfoil. Clearly all three types have very similar drag characteristics. All three types eliminate the moment and lift stall. The best pitching moment results from the thickening effect, whereas the best lift loop results from the cambering effect. It is concluded, however, that the overall best dynamic characteristics among the three types of airfoil deformation are achieved by varying the airfoil camber.

Vortex-Airfoil Interations

The problem is described in Fig. 16, which shows a point vortex of clockwise sense passing under the airfoil. Flow around a Boeing VR12 airfoil is computed for a fixed angle of attack of 5^o and a Reynolds number of 2 million. Before introducing a point vortex into the flowfield, a converged solution for an impulsively started airfoil is obtained. A point vortex of strength $-0.2v_\infty C$, where C is the airfoil chord length, is introduced at 5 chords upstream and 0.2 chord under the airfoil camber line. The vortex causes the surface pressure to fluctuate and the boundary layer to thicken. In Fig. 17, the vorticity contours show the boundary layer thickening for two locations of the passing vortex. The X-location of the passing vortex, X_Γ in the figure, is referenced to the leading edge of the airfoil and is normalized with the airfoil chord. In Fig. 18, the streamlines as well as the surface pressure are shown for different locations of the passing vortex. The effect of the passing vortex on the surface pressure appears clearly as a local suction perturbation on the lower surface where the pressure first decreases and then recovers. The pressure rise in the recovery region causes the boundary layer to thicken.

Fig. 19 shows the variation of the pressure coefficient at selected airfoil surface points with respect to the vortex location. Also shown in the figure is the pressure difference between the lower and upper surface at the same selected surface points. The large unsteady variations in the surface pressure are the main cause of BVI noise. Comparing the amplitude of the variations, it is obvious that the region close to the leading edge is a dominate source of BVI noise.

The feasibility of reducing the BVI noise by an active airfoil deformation is investigated. The front part of the airfoil is made to rotate upward as the point vortex approaches the airfoil. It then rotates back as the vortex passes under the leading edge of the airfoil. Computations have presently been limited to three locations for the center of rotation (12.5%, 25% and 37.5% of the airfoil chord) and two rotation angles (5^o and 10^o measured from the airfoil camber line). The effect of the airfoil deformation is demonstrated in Fig. 20 where the streamlines and the surface pressure are shown. In this figure, the case with the rotation centered at 25% chord and a rotation angle of 5^o is shown. This case is denoted as (.25, 5.) where the two numbers represent the rotation center and the maximum deflection angle, respectively. The front part of the airfoil starts to rotate upward at $X_\Gamma = -0.26$, reaches its maximum deflection angle at $X_\Gamma = .02$, and finally returns to its original position at $X_\Gamma = 0.29$.

In Fig. 21, the surface pressure at 2% chord is shown with respect to the X-location of the passing vortex. Also shown in the figure are the pressures for the basic airfoil and for another deformed case (.25, 10.). Both of the deformed cases show an improvement in the unsteady pressure fluctuation at the lower surface. That is, not only is the amplitude of the pressure fluctuation reduced, but the slope of the pressure rise after the point vortex has passed the leading edge is also reduced. The amplitude of the pressure fluctuation is measured by the difference between the minimum pressure and the local maximum immediately after the point vortex passes the leading edge. The two deformed cases show a reduction in the magnitude of the fluctuation by approximately the same amount, about 18% of the original value. However, the (.25, 10.) case results in a slower rate of the pressure rise. This indicates that the 10^o case is a better choice for reducing the BVI noise that originates from the lower surface of the airfoil. On the upper surface, the pressure fluctuation is aggravated by the airfoil deformation. The amplitude of the pressure fluctuation is increased by 4% of the original value for the (.25, 5.) case and by 10% for the (.25, 10.) case. Here the magnitude of the pressure fluctuation is measured by the difference between the maximum pressure and the local minimum immediately after the point vortex passes the leading edge of the airfoil.

The overall improvement achieved in these two cases can be judged by comparing the resulting lift variation with that of the basic airfoil, as shown in Fig. 22. The figure shows that the (.25, 5.) case is a better choice than the (.25, 10.) case because the magnitude of the lift variation is smaller and the slope of the lift variation is milder. Similar results were obtained when comparisons were made between (.125, 5.) and (.125, 10.) and between (.375, 5.) and (.375, 10.).

Fig. 23 shows a comparison of the computed pressure for the deformed cases (.125, 5.), (.25, 5.), and (.375, 5.) with that of the basic airfoil. At the 2% chord, the pressure variation is most improved in the (.375, 5.) case, even though this case is not much differenct from the (.25, 5.) case. However, the pressure on the upper surface in the (.375, 5.) case shows a clear advantage over the other two deformed cases. This can also be observed in the pressure difference plot. The unsteady lift is shown in Fig. 24, where it is evident that among the present cases studied, the (.375, 5.) case is the best for improving the unsteady lift variation caused by the passing vortex. The drag and the moment coefficients are also shown in this figure. It should be noted that the reduction of the pressure fluctuation is accompanied by a drag rise and a larger fluctuation of the pitching moment.

Conclusions

A numerical method based on a vorticity-velocity formulation and an integral representation for the velocity vector is extended for treating problems of multi-element airfoils, deformable airfoils, and vortex-airfoil interactions. The high efficiency of the numerical approach offers an effective way of conducting parametric studies for the demonstration of new concepts. These concepts are applied to the design of helicopter blades in order to reduce dynamic stall and BVI noise. Under the present effort, a passive slatted airfoil and an active deformable airfoil were extensively investigated.

The dynamic stall can be reduced by adding a slat or by using airfoil deformation. Three types of airfoil deformation were investigated: airfoil nose droop-down, airfoil camber change, and airfoil thickening. Even though all three are effective in reducing the dynamic stall, the most effective was found to be the camber change.

The BVI noise can be reduced by airfoil deformation. A particular type studied is the deflection-up of the airfoil's front part as a vortex passes underneath the airfoil and close to the leading edge. The amplitude as well as the rate of variation of the pressure fluctuation on the airfoil's lower surface are reduced. A best deformation case was identified under the range of parameters studied in the present work.

The current ZETA II code is capable of predicting complex unsteady flows under certain flow conditions. For larminar flows and high Reynolds number flows, the code predicts quantitatively well the aerodynamic loads and flow features. For flows of intermediate Reynolds number, where the larminar-turbulent transition is important, the code predicts well only qualitatively. The simple eddy turbulence model used here may not be adequate to treat this type of flow. Further investigations utlilzing more accurate turbulence models are needed.

Other new concepts that are potentially capable of reducing dynamic stall and BVI noise can be investigated by the present ZETA II code with minor extensions or modifications. The surface blowing/suction case, for example, can be studied by modifying the velocity boundary condition in the numerical procedure. The predicted aerodynamic characteristics with respect to a variety of parameters selected in the computation can provide a valuable guide for selecting limited parameter set for experiments. The experiments also can serve as a validation of the computer code.

References

1. Wu, J. C. "Numerical Boundary Conditions for Viscous Flow Problems", AIAA J., vol. 14, no. 8, 1976.
2. Ives, D. C., "A Modern Look at Conformal Mapping Including Multi-Connected Regions", AIAA J., vol, 14, no. 8, 1976.
3. Hall, I. M. and Suddhoo, A., "Inviscid Compressible Flow Past a Multi-Element Airfoil", AGARD-CP-365, 1984.
4. Baldwin, B. S. and Lomax, H., "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows", AIAA Paper 78-257, 1978.
5. Roach, P. J., *Computational Fluid Dynamics*, Hermosa Publishers, Albuquerque, N. M., 1972.
6. Wang, C. M., Wu, J. C. and Tuncer, I. H., "Accurate Determination of Surface Pressure in High Reynolds Number Flows", Proceedings of the First International Conference on Computational Methods in Flow Analysis, Okayama, Japan, 1988.

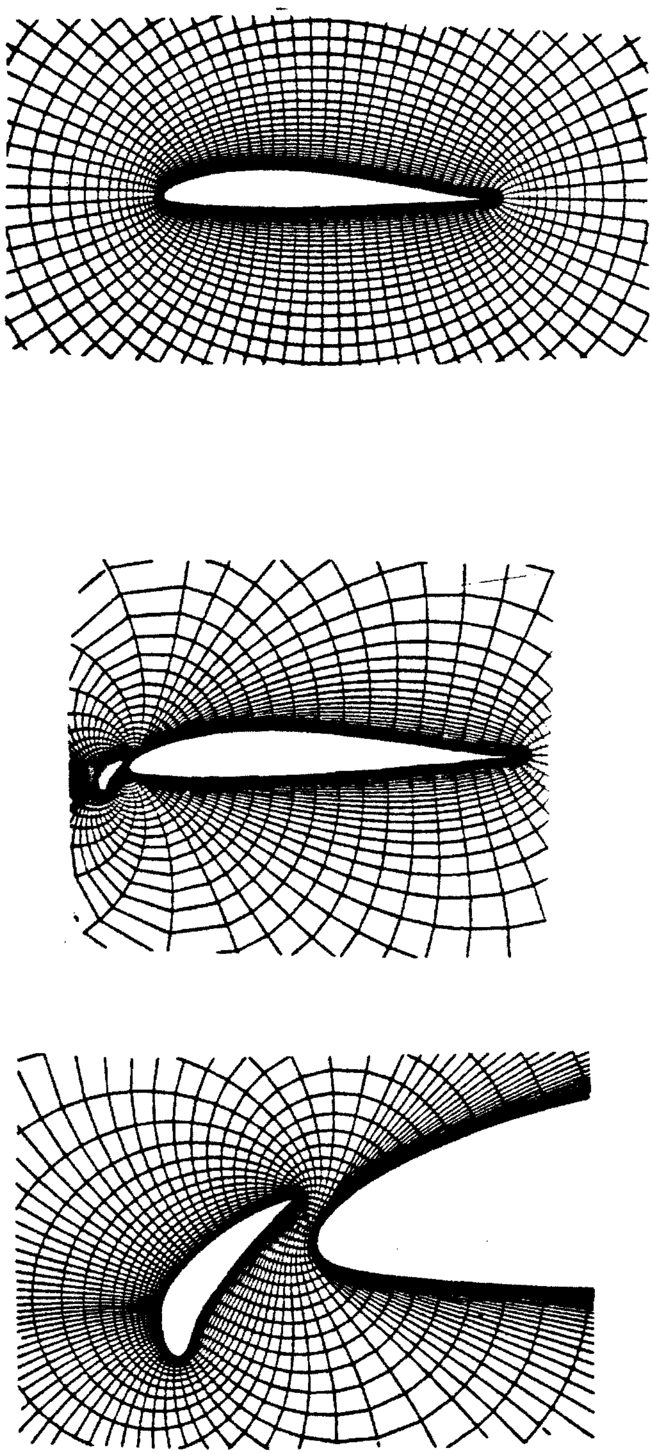


Figure 1. Grids of Basic and Slatted VR-7 Airfoil

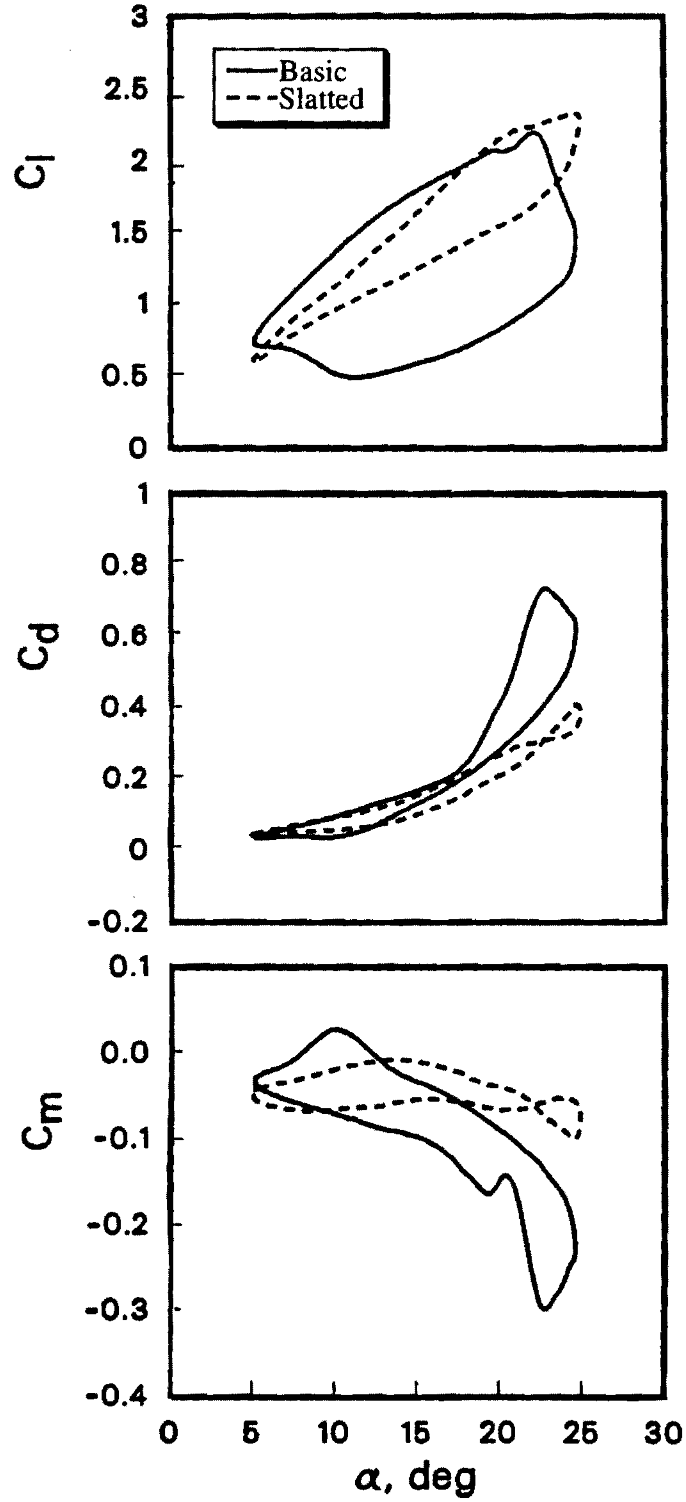


Figure 2. Experimental Load Hysteresis Loops of VR-7 Airfoil, Reynolds Number =200,000, Reduced Frequency=0.1

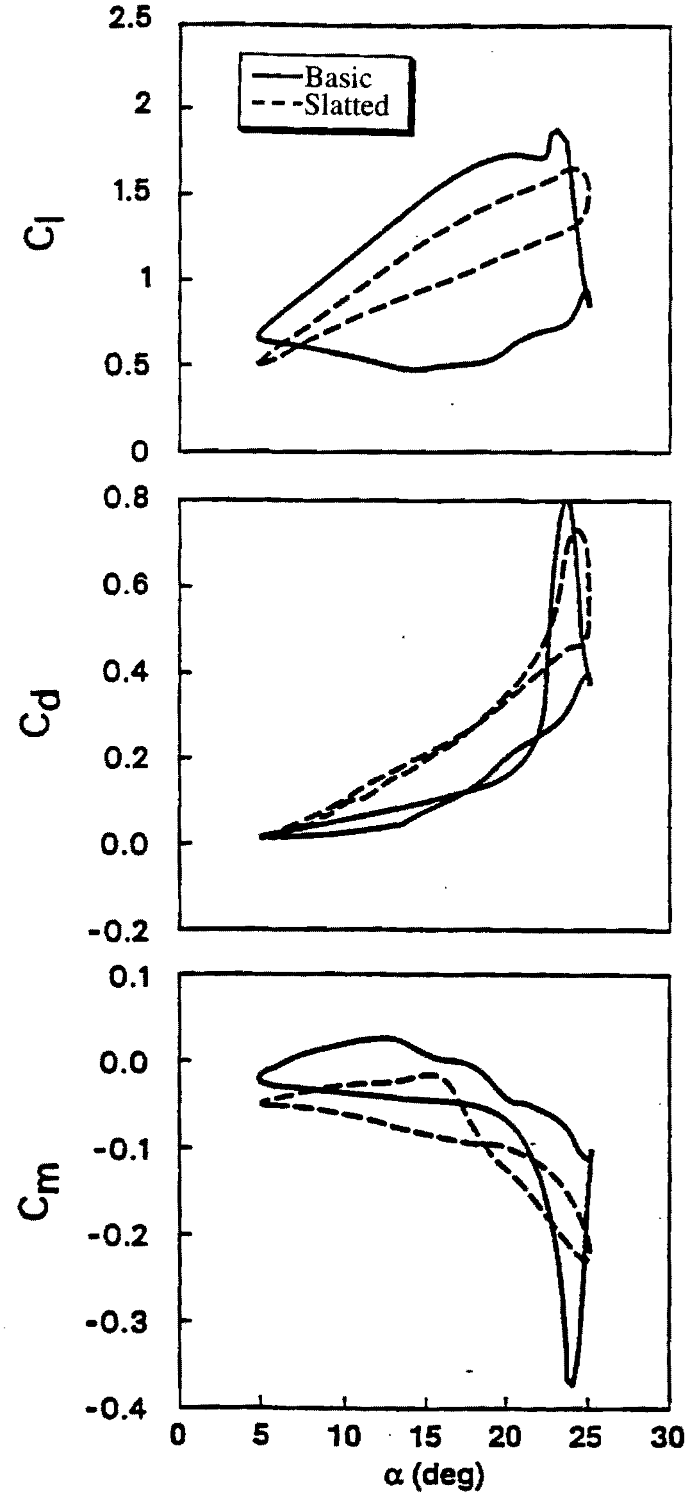


Figure 3. Computational Load Hysteresis Loops of VR-7 Airfoil, Reynolds Number =200,000, Reduced Frequency=0.1

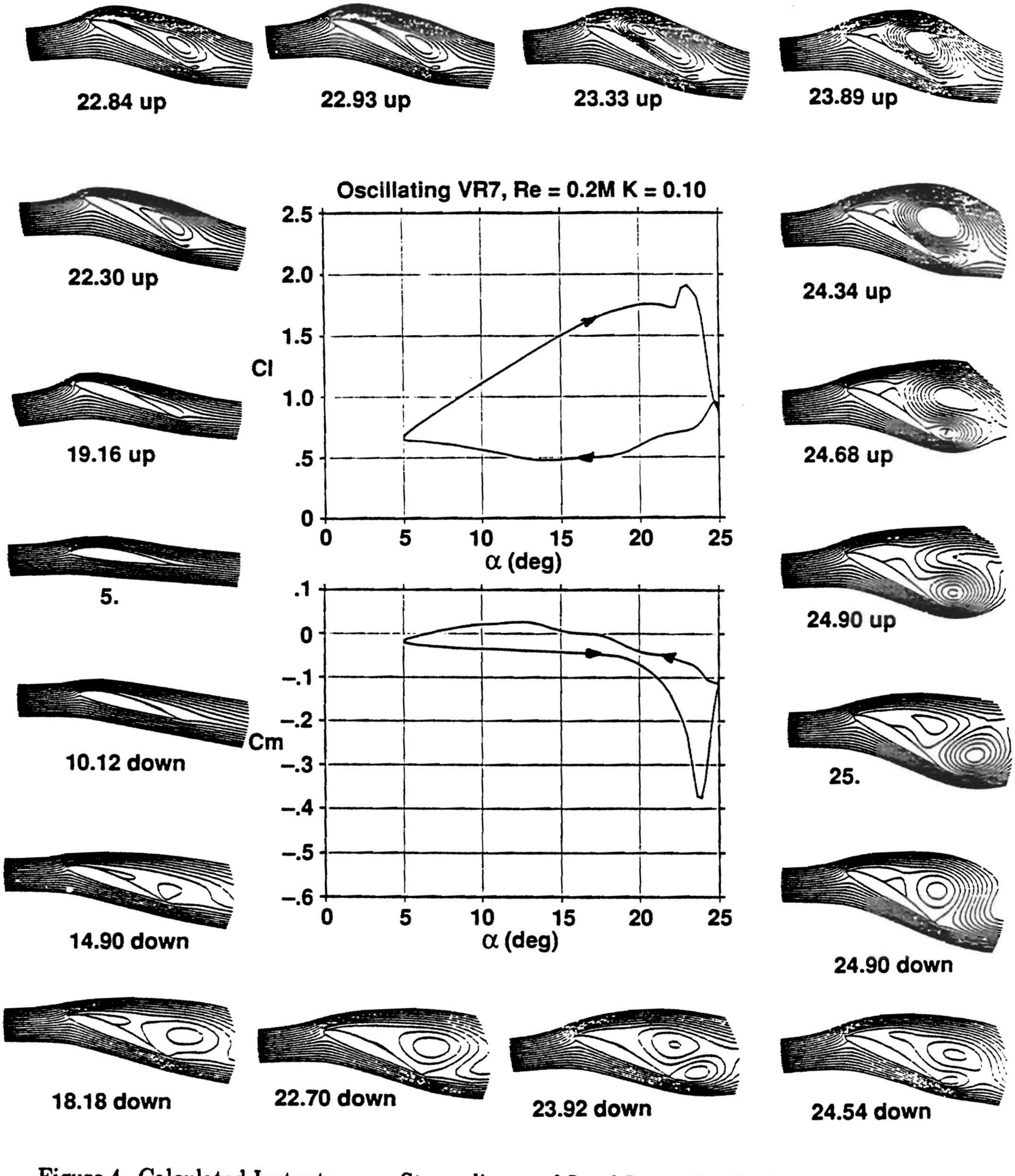


Figure 4. Calculated Instantaneous Streamlines and Load Loops for the Basic VR-7 Airfoil, Reynolds Number=200,000, Reduced Frequency=0.1

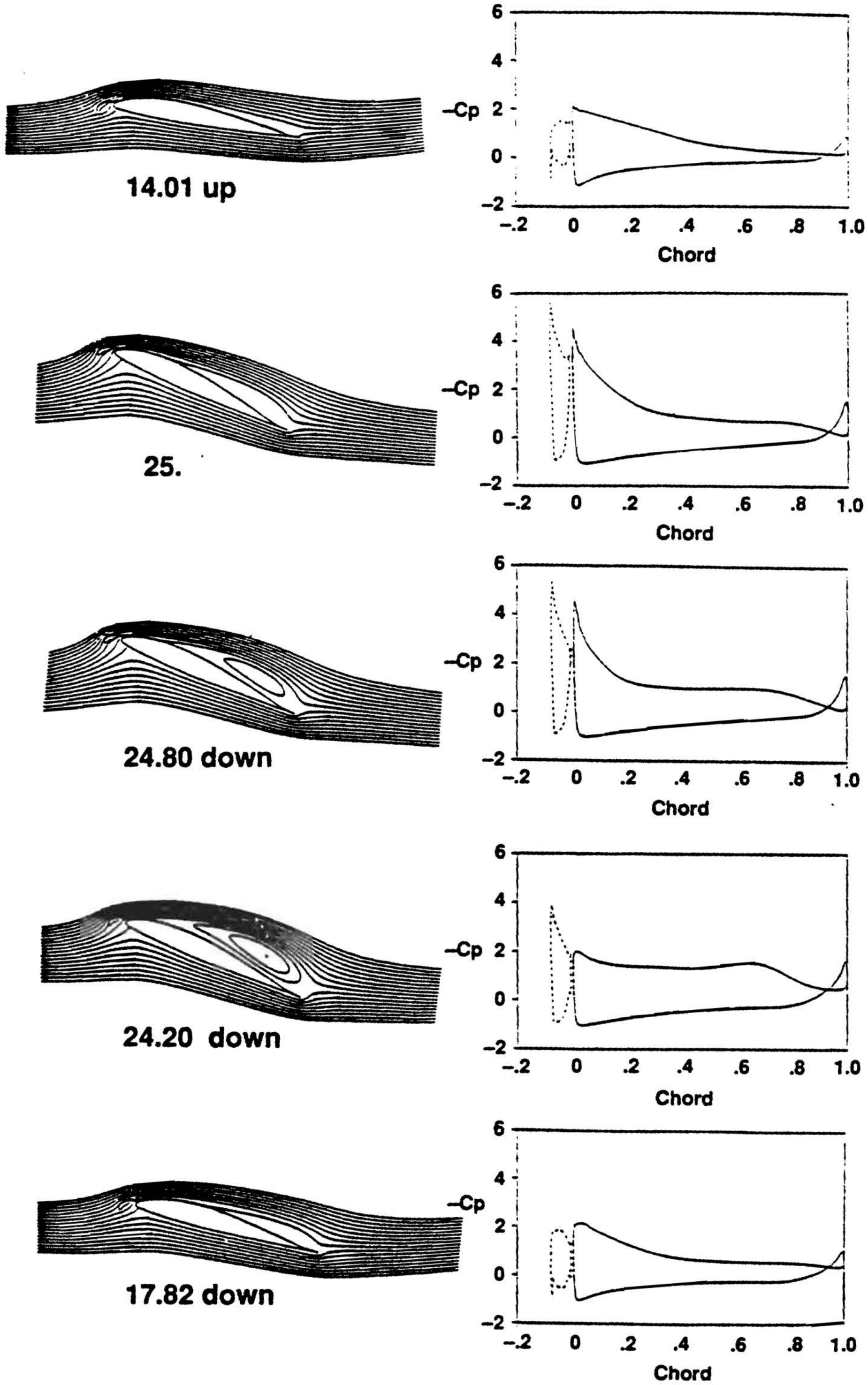


Figure 5. Calculated Instantaneous Streamlines and Surface Pressure for the Basic VR-7 Airfoil, Reynolds Number=200,000, Reduced Frequency=0.1

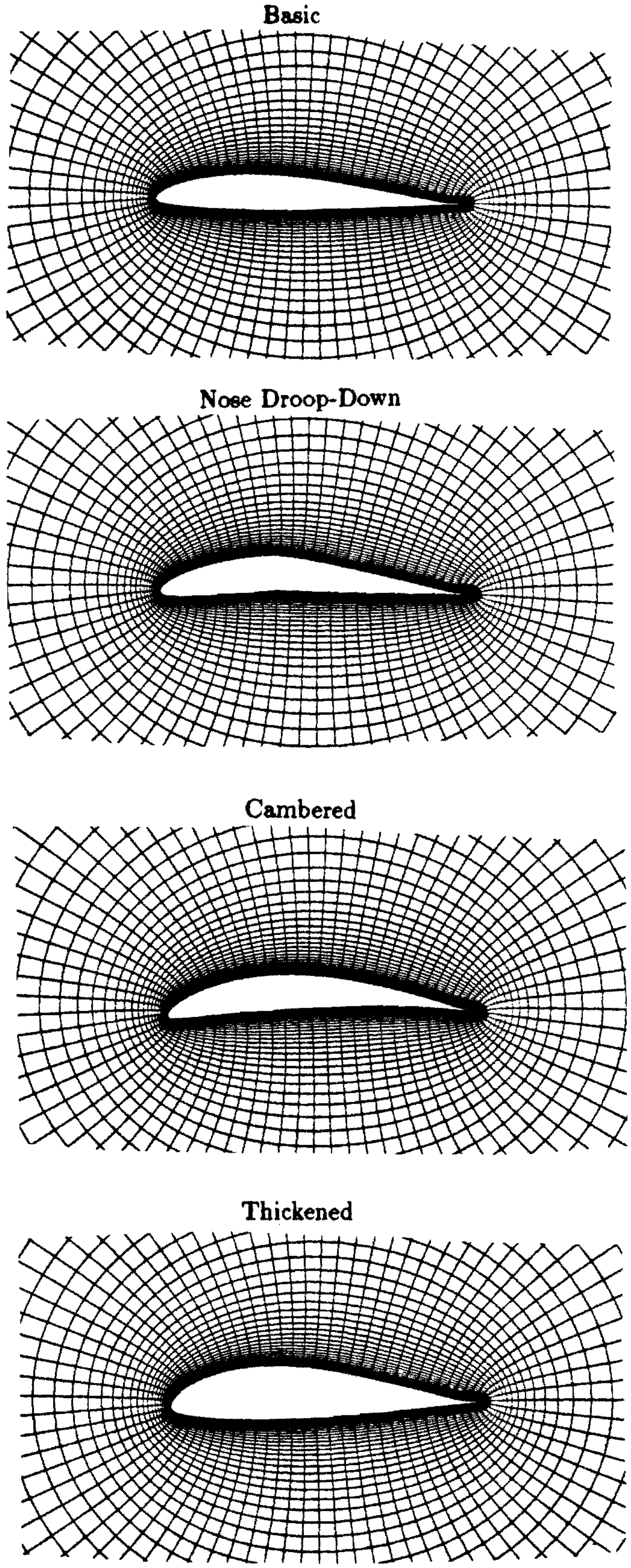


Figure 6. Types of Deformable VR-7 Airfoil

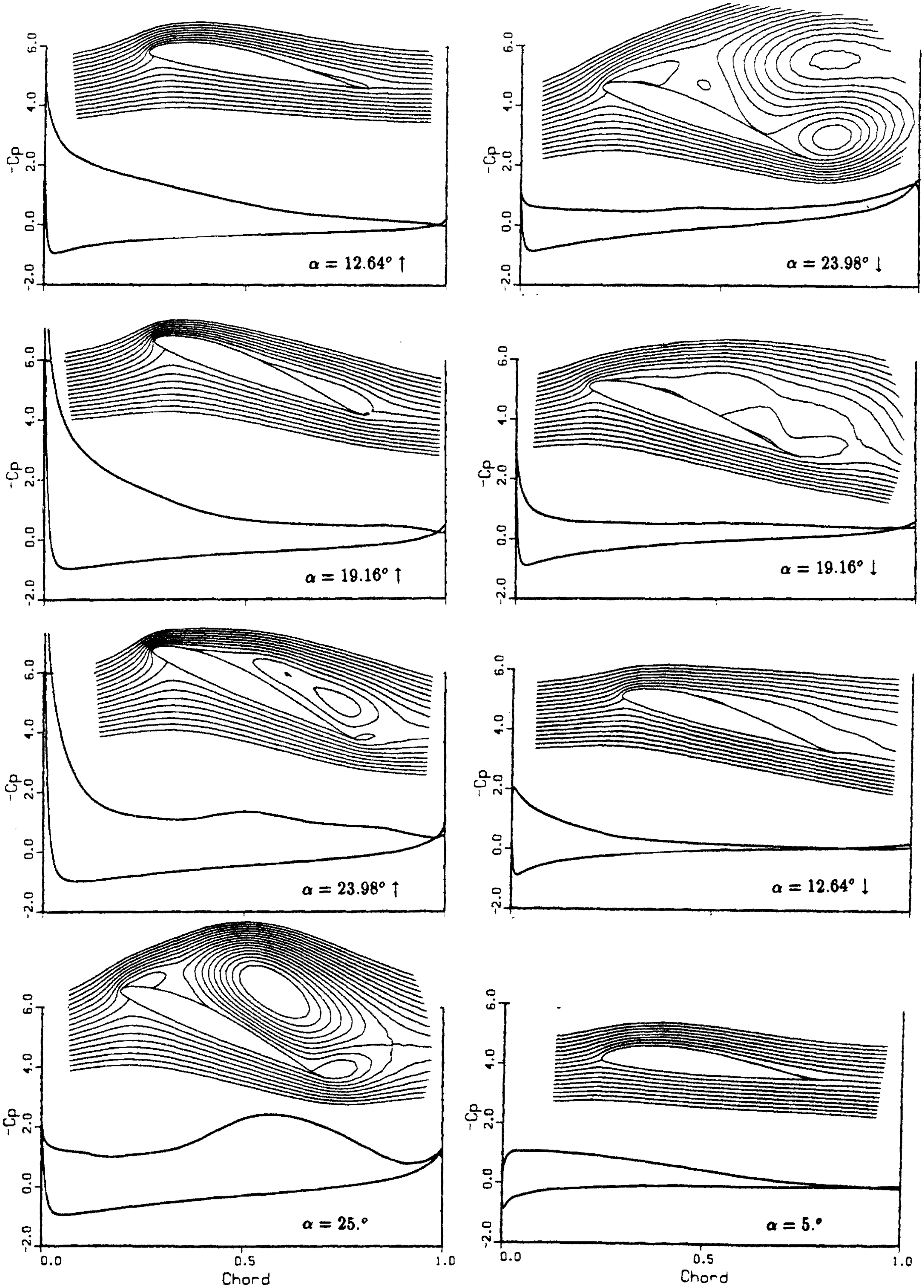


Figure 7. Calculated Instantaneous Streamlines and Surface Pressure for the Basic VR-7 Airfoil, Reynolds Number= 1 Million, Reduced Frequency=0.15

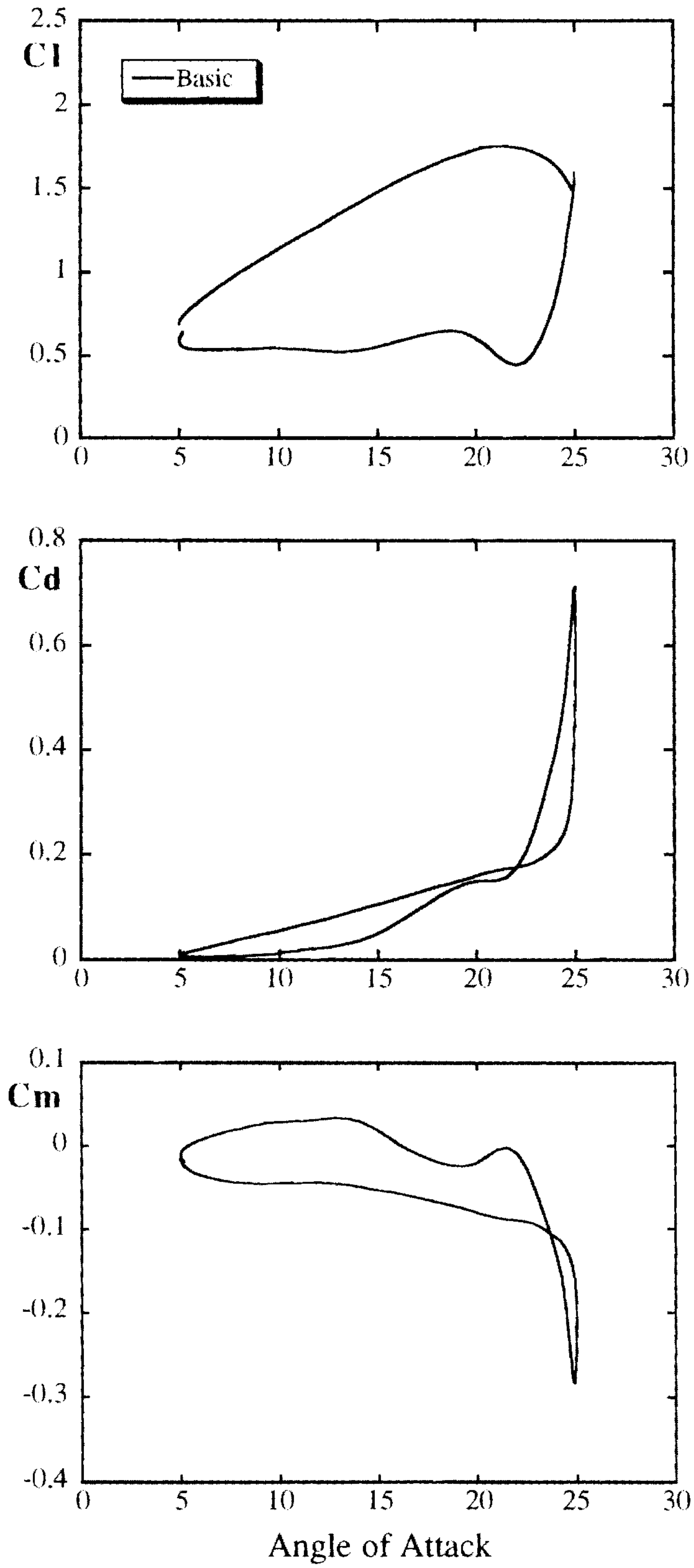


Figure 8. Calculated Loads Hysteresis Loops for the Basic VR-7 Airfoil, Reynolds Number= 1 Million, Reduced Frequency=0.15

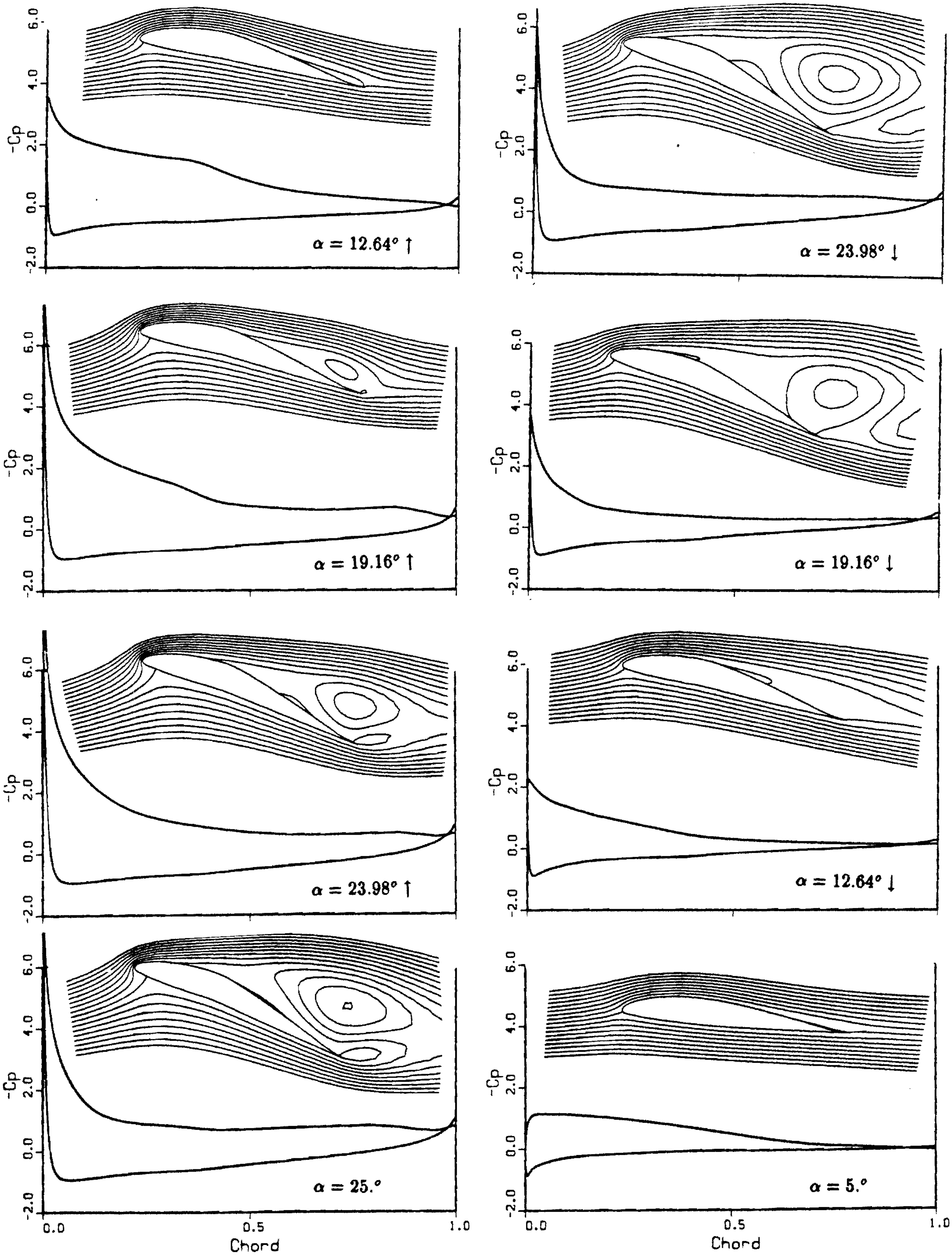


Figure 9. Calculated Instantaneous Streamlines and Surface Pressure for the Nose-Droop Down VR-7 Airfoil, Reynolds Number= 1 Million, Reduced Frequency=0.15

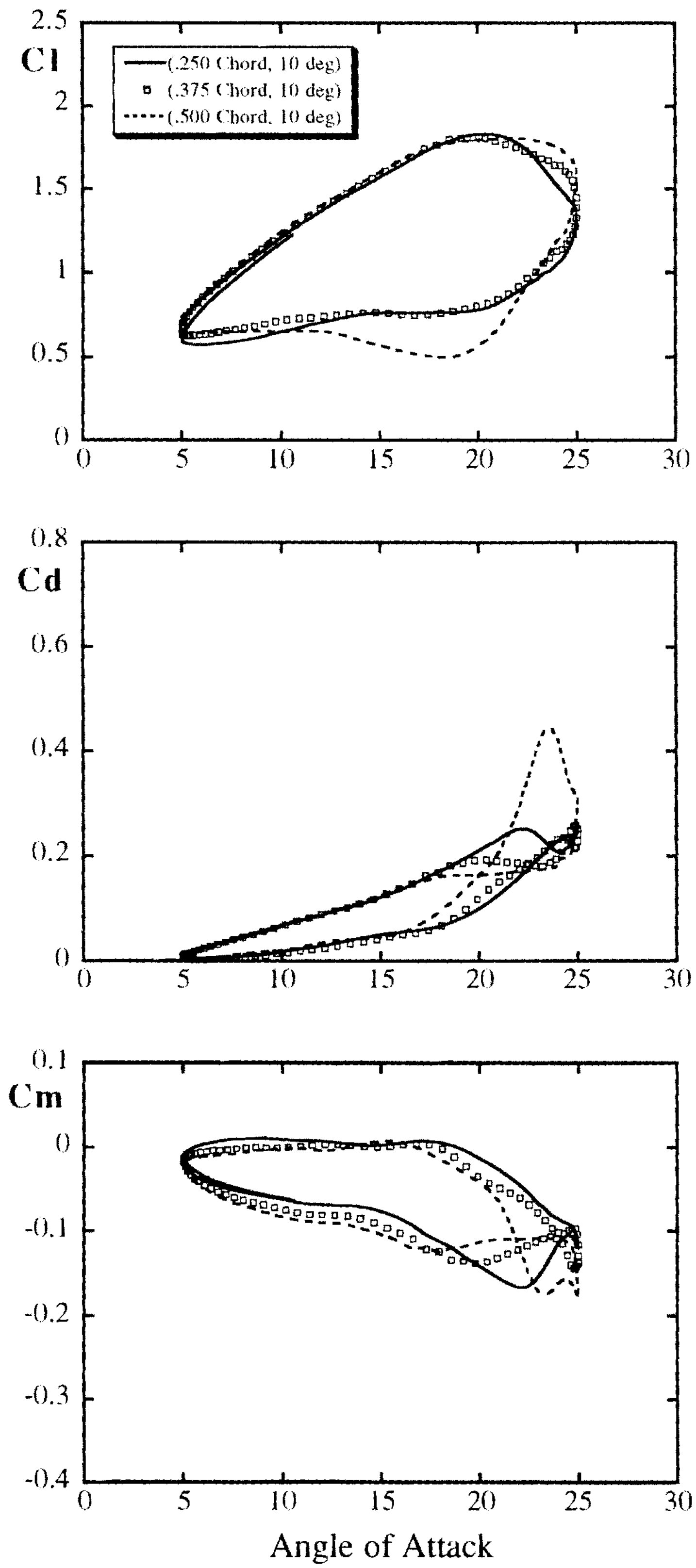


Figure 10. Calculated Loads Hysteresis Loops for the Nose-Droop Down VR-7 Airfoil, Reynolds Number= 1 Million, Reduced Frequency=0.15

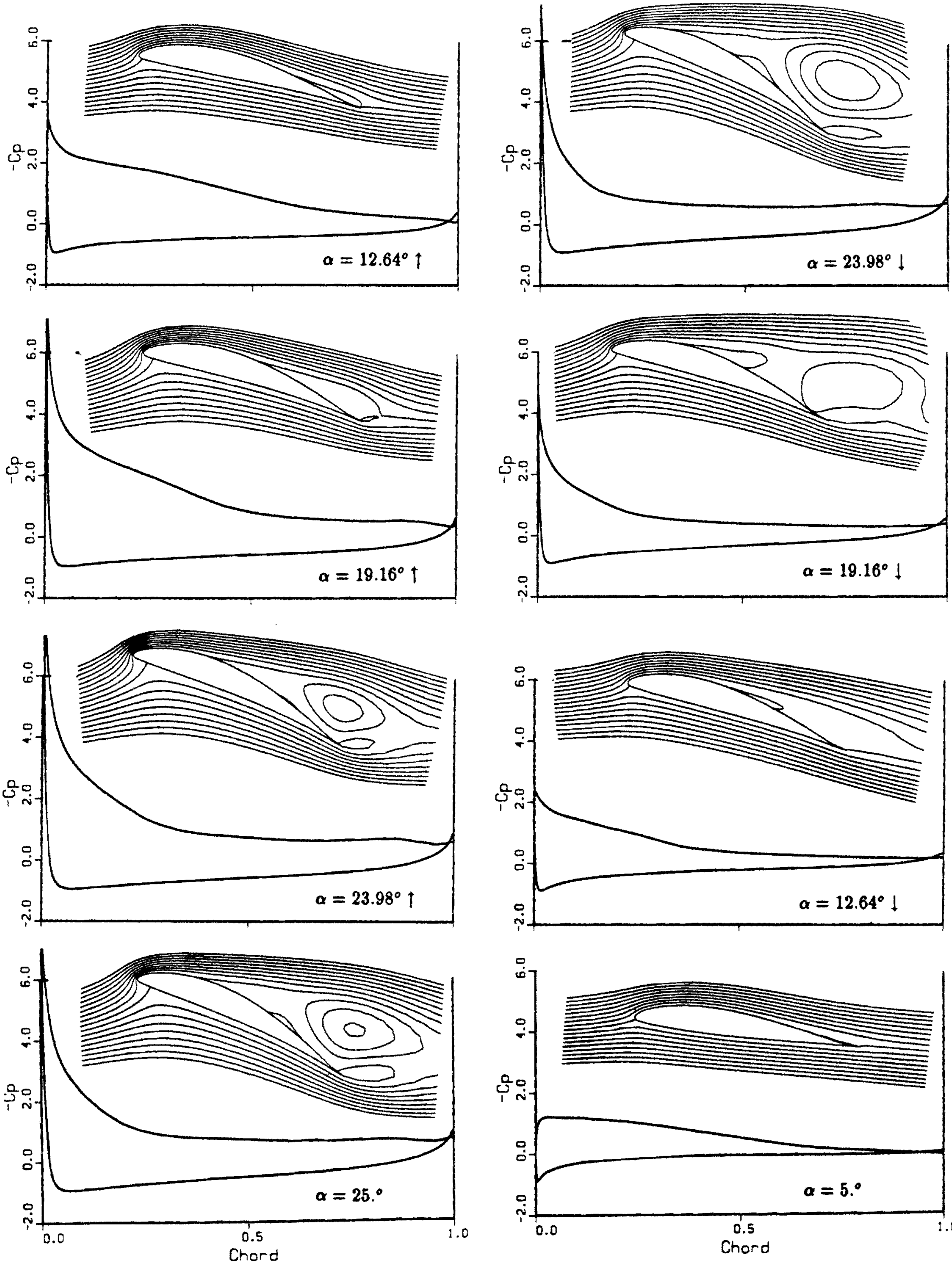


Figure 11. Calculated Instantaneous Streamlines and Surface Pressure for the Cambered VR-7 Airfoil, Reynolds Number= 1 Million, Reduced Frequency=0.15

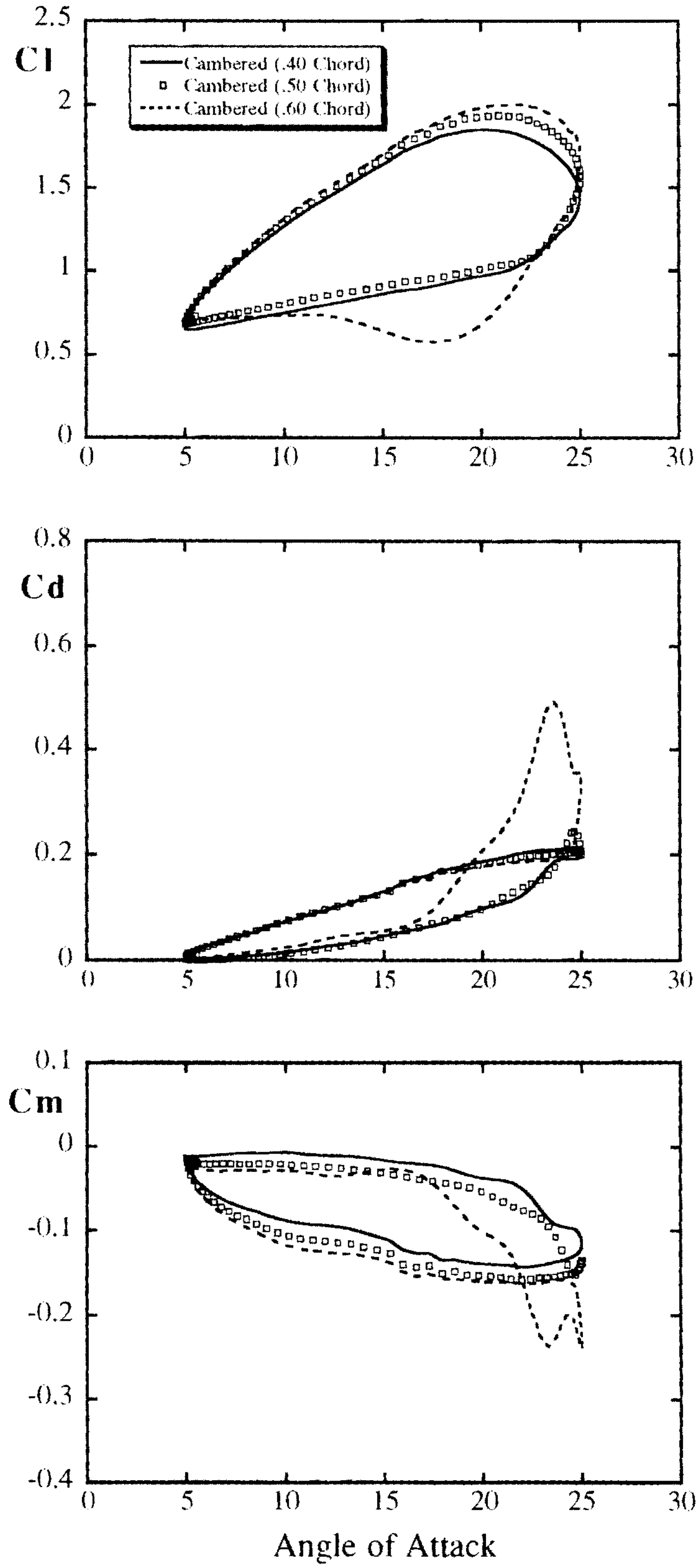


Figure 12. Calculated Loads Hysteresis Loops for the Cambered VR-7 Airfoil, Reynolds Number= 1 Million, Reduced Frequency=0.15

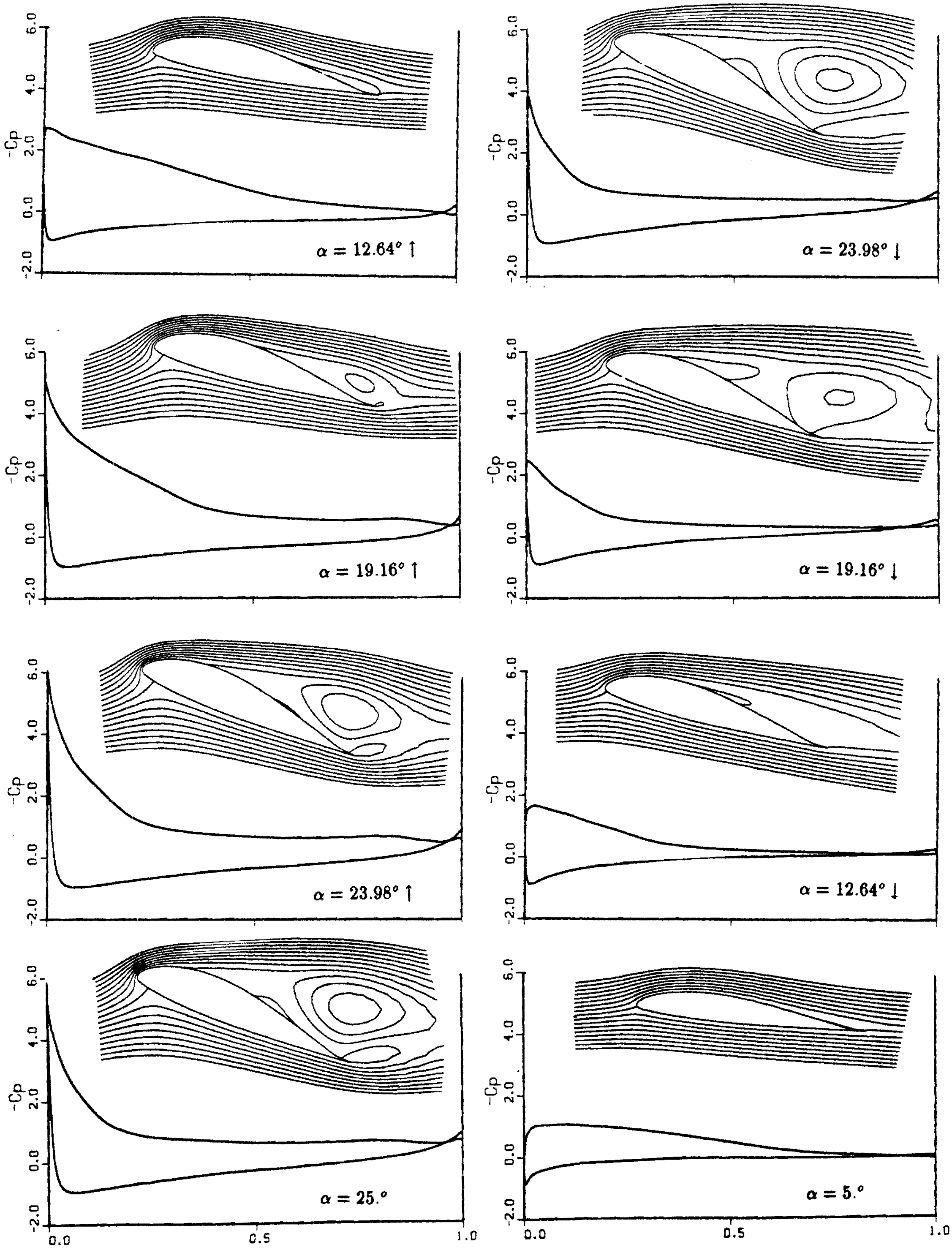


Figure 13. Calculated Instantaneous Streamlines and Surface Pressure for the Thickened VR-7 Airfoil, Reynolds Number= 1 Million, Reduced Frequency=0.15

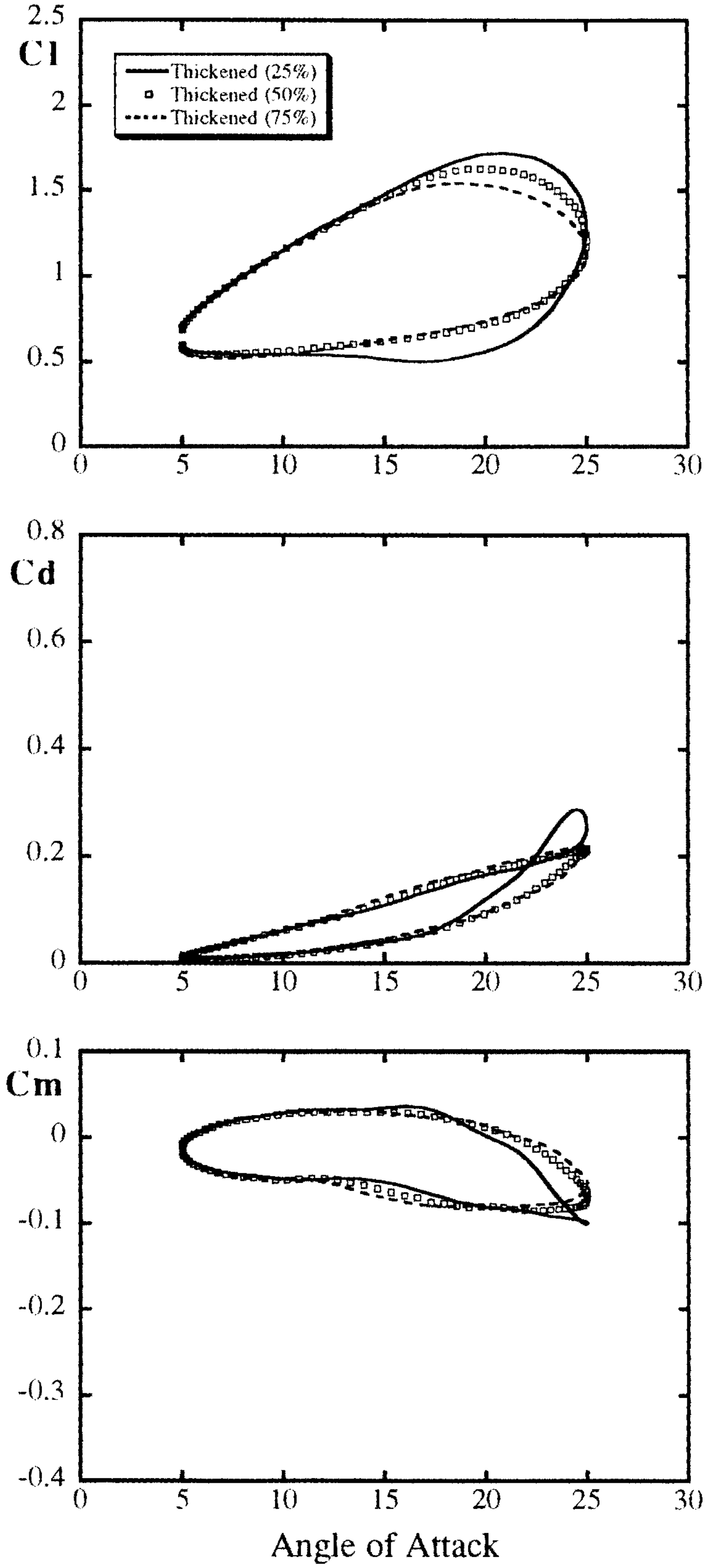


Figure 14. Calculated Loads Hysteresis Loops for the Thickened VR-7 Airfoil, Reynolds Number= 1 Million, Reduced Frequency=0.15

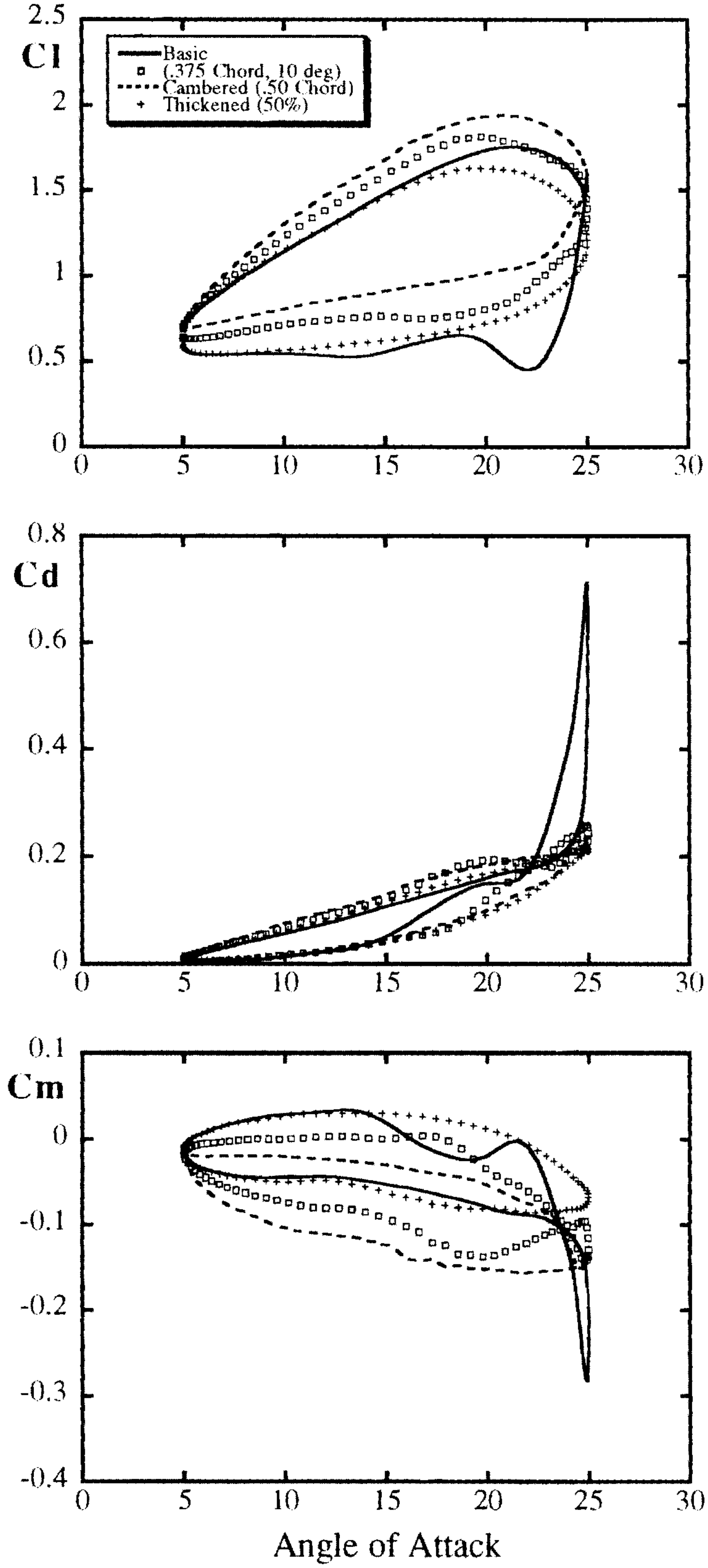


Figure 15. Calculated Comparison of Loads Hysteresis Loops for Various Deformable VR-7 Airfoil, Reynolds Number= 1 Million, Reduced Frequency=0.15

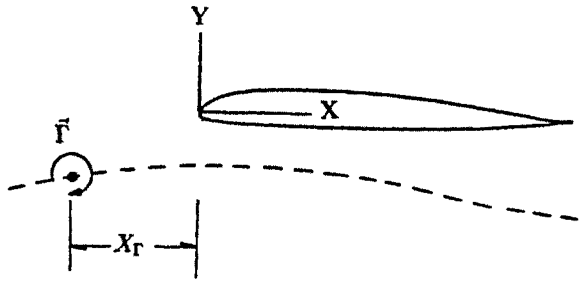


Figure 16. Schematics of Vortex-Airfoil Interaction

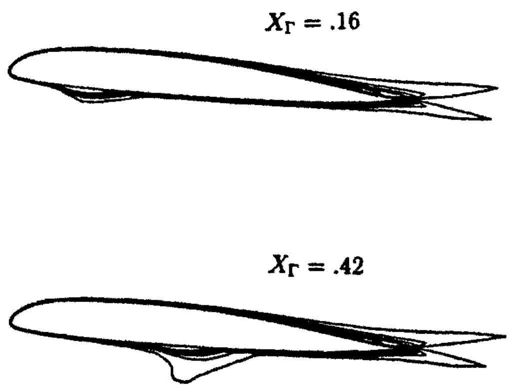


Figure 17. Distributed Vorticity Contours of Basic VR12 Airfoil

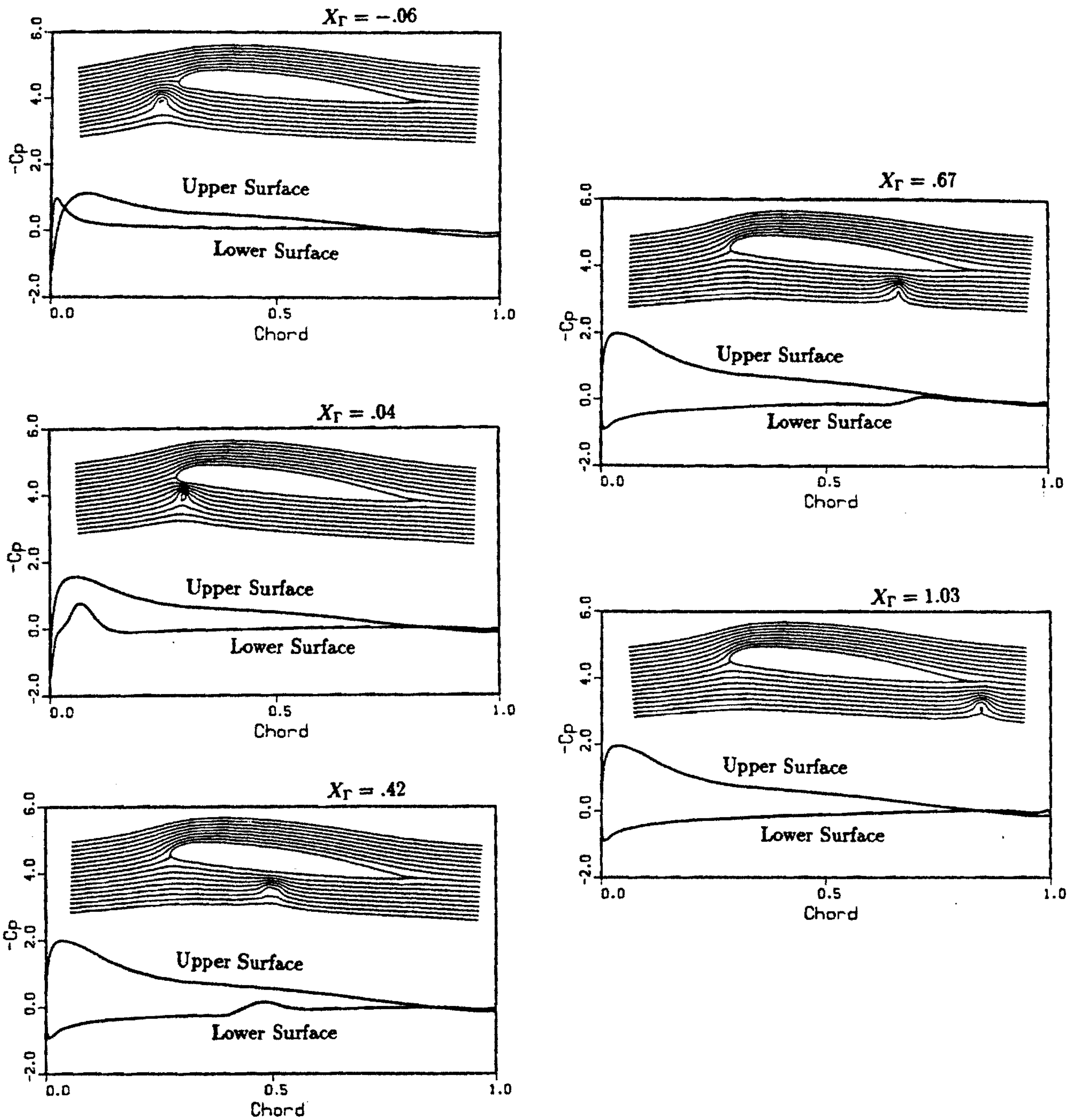


Figure 18. Streamlines and Surface Pressure of Vortex-Airfoil Interaction of Basic VR12 Airfoil

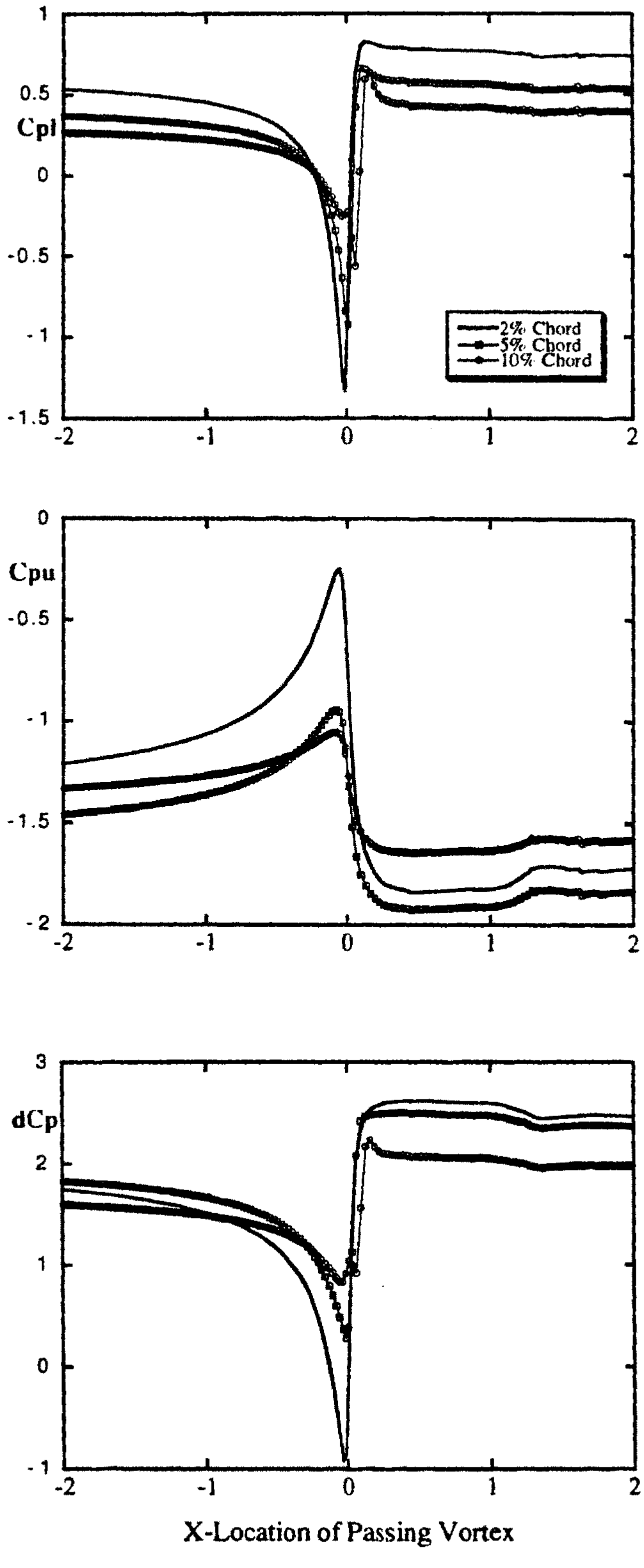


Figure 19. Surface Pressure and Pressure Difference of Basic VR12 Airfoil

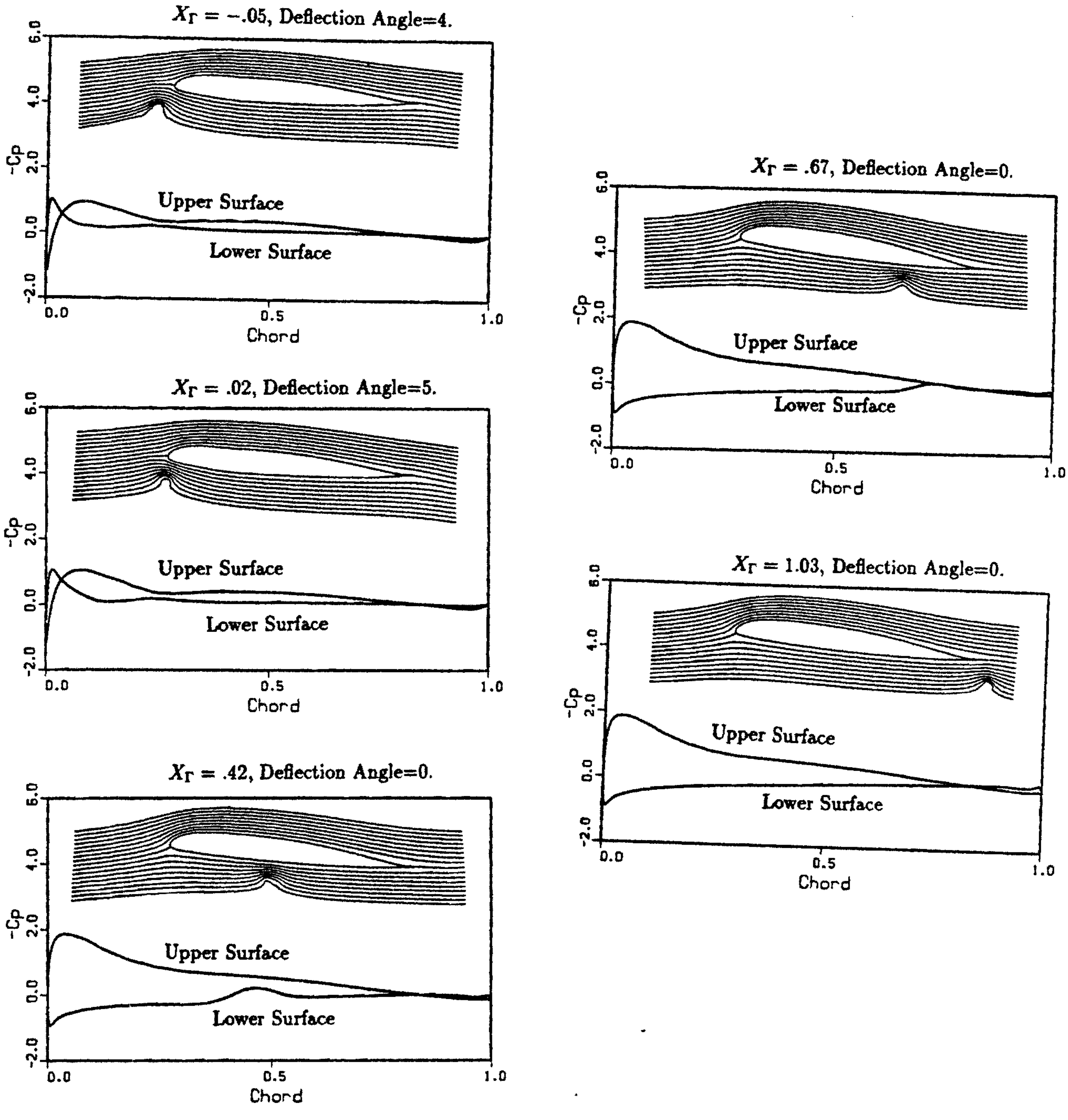


Figure 20. Streamlines and Surface Pressure of Vortex-Airfoil Interaction of (.25, 5.) Case

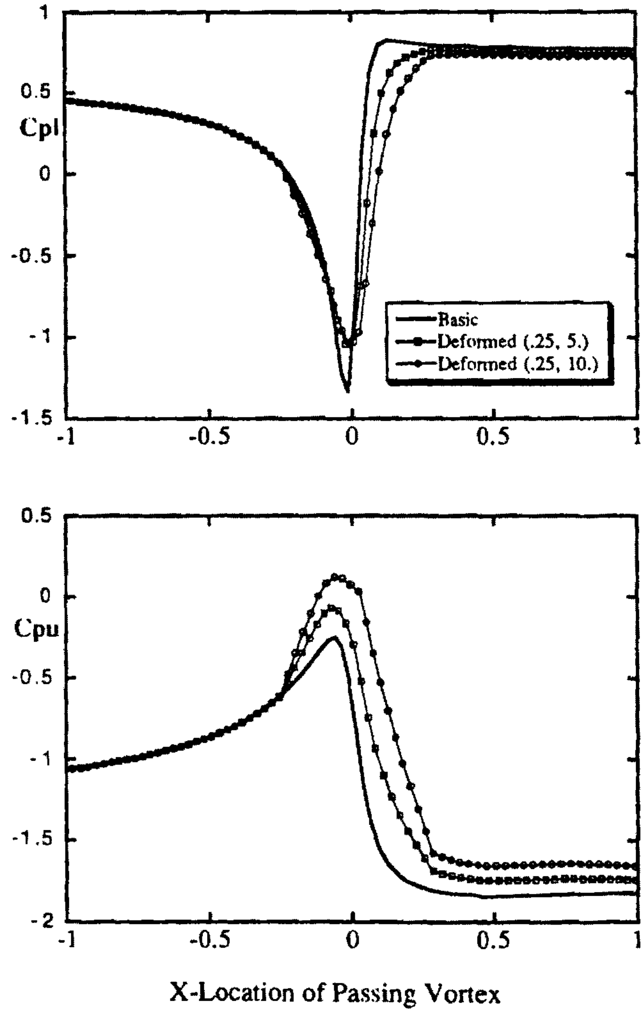


Figure 21. Comparison of Pressure Fluctuation at 2% Chord

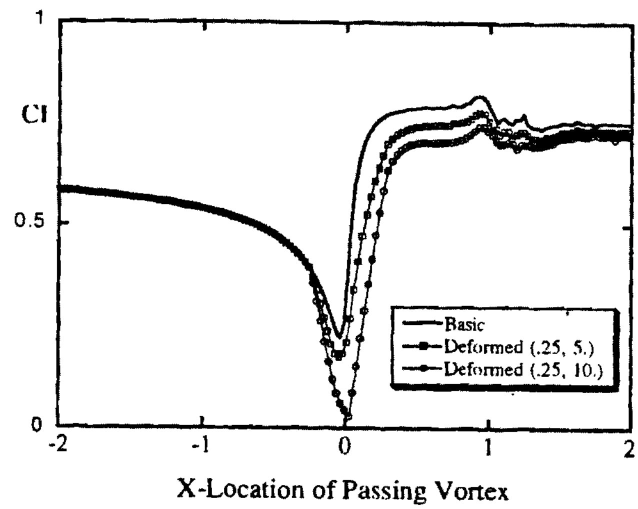


Figure 22. Comparison of Lift Coefficient

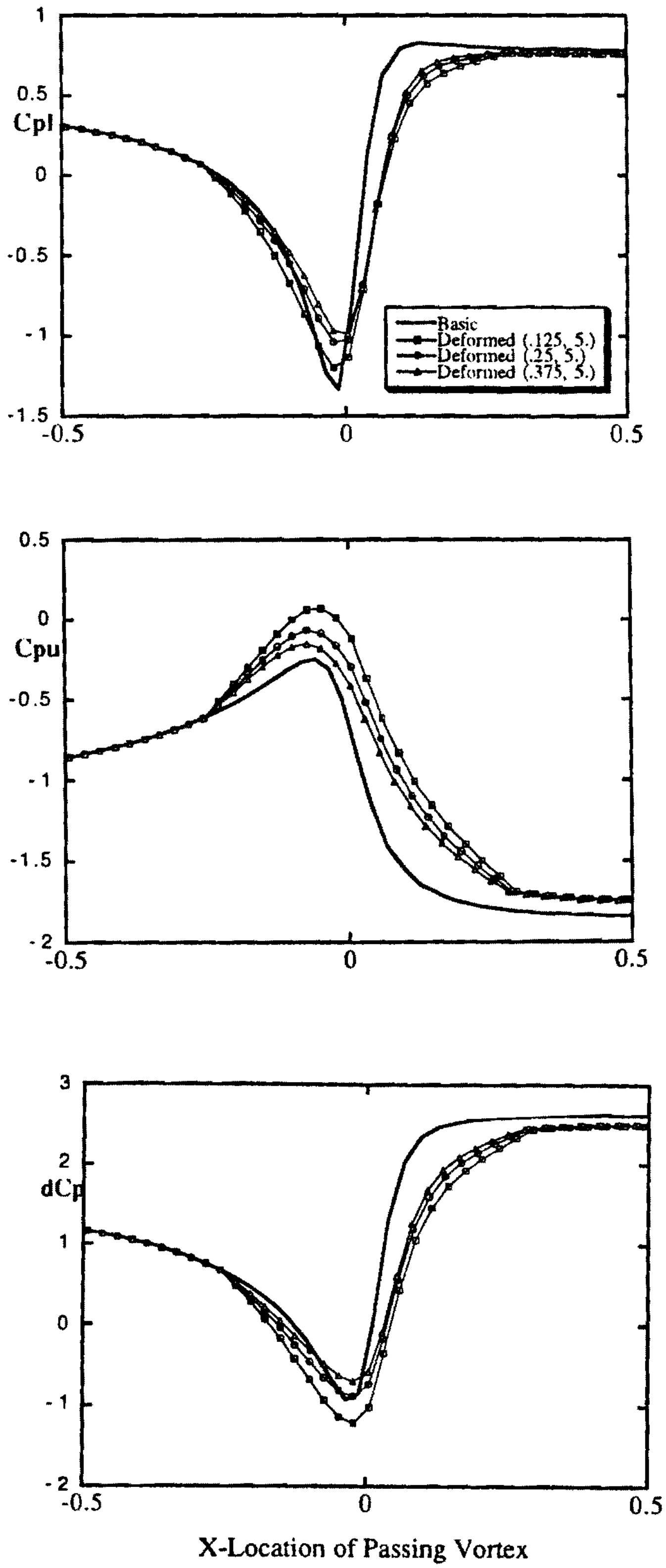


Figure 23. Comparison of Pressure Fluctuation and Pressure Difference at 2% Chord

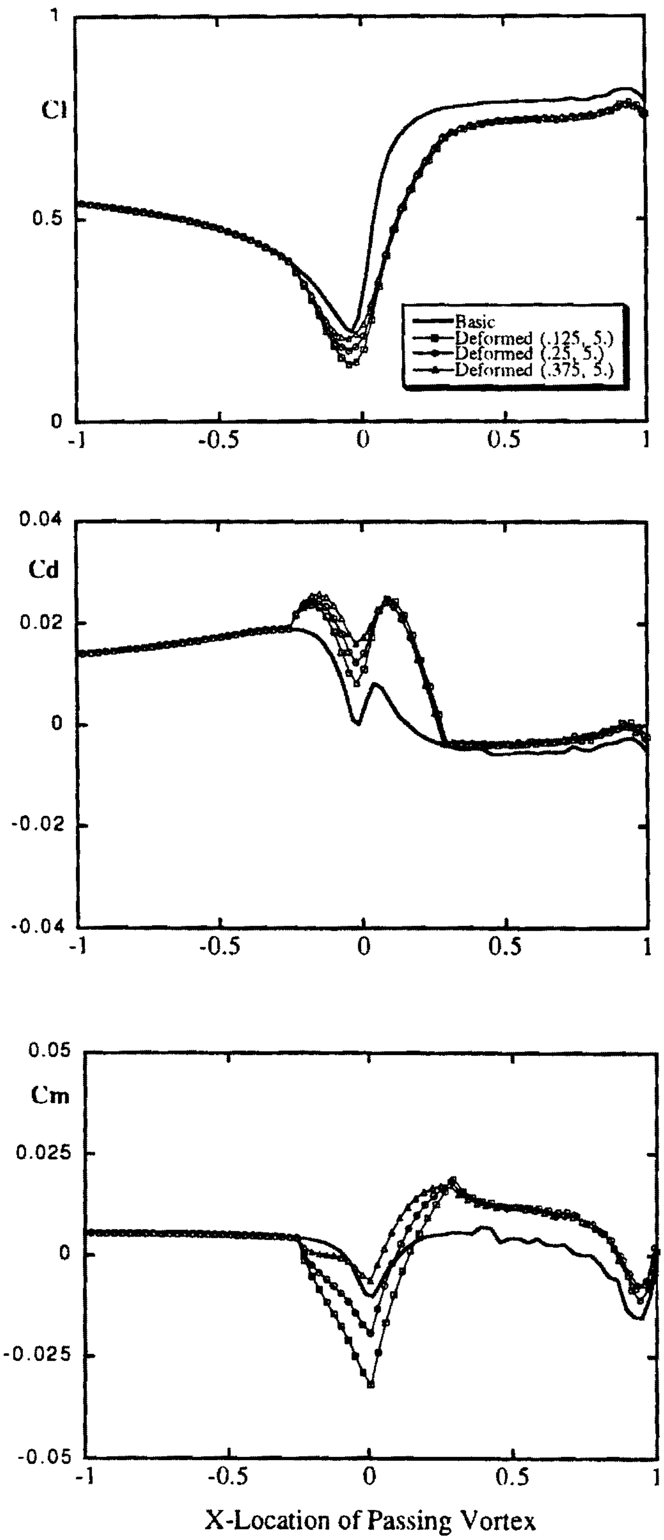


Figure 24. Comparison of Aerodynamic Loads

Appendix A

The Publications involving the present effort are as follows:

1. "A Numerical Study of Airfoil Deformation on Dynamic Stall", AIAA Paper 94-1944, 1994.
2. "Unsteady Aerodynamic Behavior of an Airfoil With and Without a Slat", Computers Fluids, vol. 22, 1993.
3. "A Numerical Study of Airfoil Deformation in Vortex-Airfoil Interaction Problems", AIAA Paper 93-3287, 1993.
4. "High Lift Concept for Rotorcraft Applications", the 49th AHS Forum, 1993.
5. "Dynamic Stall Study of a Multi-Element Airfoil", the 18th European Rotorcraft Forum, 1992.

Appendix B

A MANUAL OF ZETA II

The code ZETA II is an extension of the existing code ZETA. The program runstream, file runstream and program variables are same to that of the code ZETA. These have been included and explained in the ZETA manual. The three main programs, GEOM, ZONST, and LOADS, of the ZETA II and their input/output files are described in the following. Sample input files and output files as well as sample plots are also included.

B.1 Program GEOM

This program transforms a two-element airfiol geometry in physical plane to two concentric circles in computational plane. It generates conformal grid and information regarding the transformation, e.g., scale factor, that is necessary for flow computation by the program ZONST. The program GEOM reads in two input files: DD containing airfoil coordiantes, and GEOM.IN containing input parameters.

Input file DD contains X, Y coordinates of a slat and a main airfoil. There are MS and MA points given by a user for the coordinates of the slat and the main airfoil respectively. The MS and MA are claimed on the first statement of DD and then the slat's X, Y coordinates are followed. Each surface point is read in sequentially, starting from the trailing-edge, in a counter-clockwise order. The main airfoil's coordinates are then followed, also starting from the trailing-edge, in a clockwise order.

Input file GEOM.IN contains the following parameters:

M	Number of grid point in the circumferencial direction
N	Number of grid points in the normal direction
M3	Number of harmonics used for Theodorsen transformation
M31	Number of harmonics used for Garrick transformation
TAUS	Including angle of the slat's trailing-edge
TAUA	Including angle of the main airfoil's trailing-edge
EPSR	Real part of the half complex distance from the slat's nose to its center of curvature
EPSI	Imaginary part of the half complex distance from the slat's nose to its center of curvature
EPAR	Real part of the half complex distance from the airfoil's nose to its center of curvature
EPAI	Imaginary part of the half complex distance from the airfoil's nose to its center of curvature

MSN	The numbering of the slat nose point in file DD
MAN	The numbering of the airfoil nose point in file DD
AMP	Amplification factor of the grid stretching in the circumferencial direction, The minimum spacing is at the airfoil's trailing-edge
AMPA	Amplification factor of grid stretching in the normal direction, The minimum spacing is adjancent to the airfoil surface
AMPS	Amplification factor of grid stretching in the normal direction, The minimum spacing is adjancent to the slat surface
NA	Number of spacing in the domain wrapped around airfoil, The number of spacing for the slat region is N-NA-2
UR	An under-relaxation parameter used for Theodorsen transformation
KFC	Number of harmonics used in flow computation
LB2	Logical: =T, two-element airfoil =F, one-element airfoil

The program GEOM writes the needed geometric information on TAPE2 for flow computation by the program ZONST. It also writes necessary geometric information on TAPE3 for loads calculation by the program LOADS. For plotting of grids generated, the program writes grid coordinates on TAPE12 used by PLOT1.

B.2 Program ZONST

This is the central program of the ZETA II code. It computes flow velocity and vorticity for every time step after an initial start or after reading in a set of flow quantities that computed earlier. The program reads in the necessary geometric information stored in TAPE2, computed from the program GEOM, and an input file ZONST.IN. If a current run involves the continuation of the previous flow computation, the program ZONST also reads in the flow quantities stored in TAPE7, computed in the previous run.

Input file ZONST.IN contains the following parameters:

ALPI	Angle of Attack, in degrees
ICST	Integer denoting type of current run: =0, Initial start =1, Continuation run, steady angle of attack =3, Continuation run, rapidly pitched =4, Continuation run, oscillating sinusoidally
FR	Reduced frequency
IM	Number of spacing in the circumferencial direction
N	Number of grid lines in the normal direction

WMIN	Criterion on vorticity value to determine vortical region
DFMX	Iteration criterion for Interior vorticity computation
DRMX	Iteration criterion for surface vorticity computation
KMAX	Allowance on iteration number for surface vorticity computation
NCC	Allowance on iteration number for interior vorticity computation
URBS	Under-relaxation factor on slat's surface vorticity computation
URBA	Under-relaxation factor on airfoil's surface vorticity computation
URBP	A power factor for URBS and URBA
URR	Under-relaxation factor on interior vorticity computation
NPL	Number of time steps when the pressure is computed
NRS	Numbering of a grid line that used to count vorticity flow out of the slat region
NRA	Numbering of a grid line that used to count vorticity flow out of the airfoil region
DTI	Input time step size
DTINC	Increment of time step size between two consecutive steps
DTMAX	Maximum allowed time step size
NTMAX	Numer of time steps in one run
NTPL	Number of time steps when the flow is plotted
NTOUT	Number of time steps when the flow is printed
NTLO	Number of time steps when the loads are computed
ALPMEAN	Mean angle of attack of oscillation case
ALPAMP	Amplitude of angle of attack of oscillation case
RE	Reynolds number
IS1,IS2	Grid points on which turbulent flow starts for slat
IA1,IA2	Grid points on which turbulent flow starts for airfoil
ANUMOT	Logical: =T, new airfoil motion is initiated =F, Continuous airfoil motion of the previous run
TURB	Logical: =T, Turbulent flow computation =F, Laminar flow computation
KFC	Number of harmonics used in flow computation
LB2	Logical: =T, Two-element airfoil =F, One-element airfoil
POT	Logical: =T, Only potential flow solution is sought =F, General viscous flow solution is sought
IB1S, IB2S	Grid points where boundary layer starts for slat
IV1S, IV2S	Grid points where boundary layer ends for slat
IB1A, IB2A	Grid points where boundary layer starts for airfoil

IB1A, IB2A Grid points where boundary layer ends for airfoil

The program ZONST writes the computes flow quantities on TAPE8. These include velocity, vorticity, and other needed quantities for the continuation of the next run. In that case, the quantities on TAPE8 are copied to TAPE7. The TAPE7 will be read in by the program ZONST for the next run. The quantities on TAPE8 are:

NT	Integer denoting time level
N	Number of grid lines in the normal direction
KINS	Array contains grid number in normal direction for vortical region in slat domain
KINA	Array contains grid number in normal direction for vortical region in airfoil domain
T	Time level
DT	Time step size
VOR	Array contains vorticity
U	Array contains velocity component in the circumferencial direction
V	Array contains velocity component in the normal direction
VORLS	Integrated vorticity value flowing out of the domain enclosed by grid line of NRS
VORLA	Integrated vorticity value flowing out of the domain enclosed by grid line of NRA
GAMAS	Integrated vorticity value flow to the singular domain
AGA1_S	Cosine Fourier coefficients for the surface vorticity of the slat
BGA1_S	Sine Fourier coefficients for the surface vorticity of the slat
AGA1_A	Cosine Fourier coefficients for the surface vorticity of the airfoil
BGA1_A	Sine Fourier coefficients for the surface vorticity of the airfoil

The program also writes vorticity and instantaneous stream function values on TAPE9 for flow plots by the program PLOT2.

B.3 Program LOADS

This program computes surface pressure using an integral representation for the pressure. It also integrates the surface pressure and the viscous stress to get aerodynamic loads. The program reads in input file ZONST.IN, TAPE3, and TAPE7. It writes out the aerodynamic loads on TAPE 32 and TAPE33 for the slat and the airfoil respectively. The program writes the surface pressure on TAPE4 for pressure plots by the program PLOT3.

B.4 Sample Inputs

```
---------------------------------------
|                                     |
|           INPUT FILE   DD           |
|                                     |
---------------------------------------

  43 79                             | MS   MA
-.00956   .02352               ---  | X    Y   of Slat, Trailing-edge first
-.02525   .00853                    |   Points read in counter-clockwise
-.03223   .00136                    |
-.03889  -.00605
-.04509  -.01345
-.05108  -.02089
-.05689  -.02828
-.05882  -.03102
-.06066  -.03422
-.06241  -.03794
-.0642   -.04252
-.06493  -.04456
-.06574  -.04679
-.06675  -.04918
-.06737  -.05061
-.06826  -.05245
-.06882  -.05329
-.06955  -.05399
-.07048  -.05452
-.07169  -.05484
-.07287  -.05479
-.074    -.05447
-.07477  -.05397
-.07547  -.05324
-.0761   -.05239
-.07729  -.05039
-.07876  -.04666
-.07988  -.04149
-.08029  -.03842
-.08045  -.03509
-.07999  -.03162
-.07935  -.02856
-.07813  -.02544
-.07664  -.02202
-.07491  -.01906
-.07254  -.01583
-.06525  -.0077
-.05715  -.00041
-.04851   .00595
-.03914   .01178
-.02955   .01661
-.0199    .02025
-.00956   .02352
  1.  0.                        ----  | X    Y   of Airfoil, Trailing-edge first
 .99   -.0011                         |   Points read in clockwise
 .97   -.003
 .95   -.0044
 .925  -.00609
 .9    -.00744
 .85   -.0105
 .8    -.01346
 .75   -.01639
 .7    -.01929
 .65   -.02207
 .6    -.02464
 .55   -.02712
 .5    -.02952
```

.45	-.03148
.4	-.03271
.35	-.03308
.3	-.03273
.25	-.03186
.2	-.03055
.15	-.02855
.125	-.02709
.1	-.02516
.08	-.0232
.065	-.02154
.05	-.01966
.035	-.01751
.025	-.01575
.02	-.01467
.016	-.01367
.0125	-.01263
.01	-.01172
.008	-.01086
.0065	-.0101
.005	-.00919
.0035	-.008
.002	-.00633
.001	-.0046
.0005	-.0033
0.	0.
.0005	.00337
.001	.00483
.002	.00696
.0035	.00943
.005	.01149
.0065	.0133
.008	.01494
.01	.01695
.0125	.01923
.016	.02213
.02	.02512
.025	.02846
.035	.03423
.05	.04144
.065	.04759
.08	.05299
.1	.05922
.125	.06565
.15	.07091
.2	.07887
.25	.08378
.3	.08592
.35	.08574
.4	.08365
.45	.07984
.5	.07451
.55	.06781
.6	.05996
.65	.05171
.7	.04322
.75	.03442
.8	.02527
.85	.01575
.9	.00558
.925	.00117
.95	.0005
.97	.0003
.99	.0001
1.	0.

```
 -----------------------------------------
|                                         |
|        INPUT FILE  GEOM.IN              |
|                                         |
 -----------------------------------------

121  51  40  40                    |  M     N     M3    M31
0.3  0.02                          |  TAUS   TAUA
0.0  -0.0072  -0.0216   0.         |  EPSR   EPSI   EPAR   EPAI
24   40                            |  MSN   MAN
1.04  1.07   1.18                  |  AMP   AMPA   AMPS
29   1.   160    T                 |  NA    UR    KFÇ   LB2
```

```
----------------------------------------
|                                      |
|        INPUT FILE  ZONST.IN          |
|                                      |
----------------------------------------

15.  0  .10  120   51           |  ALPI   ICST   FR   IM   N
.001 .001 .00003 60  45         |  WMIN   DFMAX   DRMX   KMAX   NCC
.1   .1   .3   .8               |  URBS   URBA   URBB   URR
0  12  15                       |  NTPL   NRS   NRA
.002   0.000 .9   20            |  DTI   DTINC   DTMAX   NTMAX
0  0  0                         |  NTPL   NTOUT   NTLO
15.  10.                        |  ALPMEAN    ALPAMP
1000000.                        |  RE
29 46 22 45                     |  IS1   IS2   IA1   IA2
F  T                            |  ANUMOT   TURB
160                             |  KFC
T                               |  LB2 (=T if two elemet airfoil)
F                               |  POT (=T if only potential flow computed)
35   66   80   120              |  IB1S    IV1S    IV2S    IB2S
 5   40   70   115              |  IV1A    IB1A    IB2A    IV2A
```

B.5 Sample Outputs

```
-------------------------------------------
|                                         |
|          OUTPUT FROM   GEOM             |
|                                         |
-------------------------------------------
```

** AIRFORIL TRAILING EDGE ANGLE = 1.3317016595227E-2
** SLAT TRAILING EDGE ANGLE = 4.264711851645
RAP RA 0.7526887384186, 0.7358689704892

***** AFTER TRANSFORMATION ******

	AIRFOIL		SLAT	
I	XA	YA,	XS	YS
1	3.6000	0.0000	-0.2882	-0.1137
2	3.5904	-0.0011	-0.2887	-0.1166
3	3.5618	-0.0043	-0.2891	-0.1196
4	3.5146	-0.0091	-0.2895	-0.1226
5	3.4481	-0.0144	-0.2897	-0.1258
6	3.3621	-0.0203	-0.2897	-0.1291
7	3.2573	-0.0266	-0.2897	-0.1325
8	3.1349	-0.0338	-0.2896	-0.1360
9	2.9979	-0.0423	-0.2893	-0.1397
10	2.8491	-0.0511	-0.2889	-0.1434
11	2.6911	-0.0604	-0.2884	-0.1473
12	2.5275	-0.0698	-0.2877	-0.1513
13	2.3612	-0.0791	-0.2870	-0.1553
14	2.1950	-0.0877	-0.2861	-0.1593
15	2.0312	-0.0959	-0.2851	-0.1633
16	1.8723	-0.1036	-0.2840	-0.1673
17	1.7204	-0.1102	-0.2827	-0.1712
18	1.5764	-0.1151	-0.2813	-0.1751
19	1.4408	-0.1182	-0.2797	-0.1788
20	1.3134	-0.1195	-0.2780	-0.1823
21	1.1943	-0.1195	-0.2761	-0.1856
22	1.0834	-0.1185	-0.2741	-0.1886
23	0.9806	-0.1169	-0.2720	-0.1913
24	0.8857	-0.1150	-0.2698	-0.1936
25	0.7987	-0.1128	-0.2674	-0.1954
26	0.7190	-0.1105	-0.2648	-0.1967
27	0.6464	-0.1079	-0.2620	-0.1974
28	0.5803	-0.1051	-0.2591	-0.1976
29	0.5202	-0.1021	-0.2562	-0.1972
30	0.4656	-0.0989	-0.2532	-0.1962
31	0.4160	-0.0955	-0.2503	-0.1944
32	0.3709	-0.0919	-0.2476	-0.1918
33	0.3298	-0.0882	-0.2450	-0.1880
34	0.2924	-0.0845	-0.2424	-0.1831
35	0.2584	-0.0808	-0.2397	-0.1769
36	0.2275	-0.0773	-0.2368	-0.1695
37	0.1995	-0.0738	-0.2336	-0.1608
38	0.1741	-0.0705	-0.2300	-0.1509
39	0.1511	-0.0673	-0.2257	-0.1397
40	0.1304	-0.0642	-0.2203	-0.1274
41	0.1118	-0.0611	-0.2136	-0.1146
42	0.0951	-0.0581	-0.2055	-0.1020
43	0.0802	-0.0551	-0.1967	-0.0903
44	0.0671	-0.0520	-0.1879	-0.0793
45	0.0554	-0.0489	-0.1793	-0.0687
46	0.0453	-0.0457	-0.1709	-0.0583
47	0.0364	-0.0424	-0.1627	-0.0482
48	0.0288	-0.0390	-0.1545	-0.0384
49	0.0223	-0.0353	-0.1465	-0.0288
50	0.0169	-0.0315	-0.1385	-0.0196
51	0.0123	-0.0275	-0.1308	-0.0108
52	0.0086	-0.0234	-0.1232	-0.0024

53	0.0056	-0.0191	-0.1157	0.0056
54	0.0034	-0.0147	-0.1084	0.0131
55	0.0018	-0.0102	-0.1015	0.0202
56	0.0009	-0.0055	-0.0948	0.0270
57	0.0005	-0.0008	-0.0885	0.0334
58	0.0008	0.0040	-0.0824	0.0398
59	0.0015	0.0088	-0.0765	0.0459
60	0.0029	0.0138	-0.0707	0.0519
61	0.0048	0.0189	-0.0650	0.0577
62	0.0071	0.0239	-0.0596	0.0630
63	0.0098	0.0287	-0.0547	0.0677
64	0.0128	0.0333	-0.0502	0.0718
65	0.0160	0.0378	-0.0463	0.0753
66	0.0195	0.0422	-0.0428	0.0782
67	0.0232	0.0466	-0.0399	0.0806
68	0.0271	0.0510	-0.0375	0.0824
69	0.0313	0.0554	-0.0358	0.0837
70	0.0354	0.0594	-0.0348	0.0844
71	0.0399	0.0637	-0.0344	0.0847
72	0.0455	0.0687	-0.0350	0.0845
73	0.0518	0.0741	-0.0367	0.0839
74	0.0581	0.0791	-0.0390	0.0832
75	0.0650	0.0844	-0.0422	0.0823
76	0.0725	0.0898	-0.0460	0.0811
77	0.0808	0.0956	-0.0506	0.0797
78	0.0900	0.1016	-0.0559	0.0781
79	0.1002	0.1079	-0.0618	0.0763
80	0.1116	0.1145	-0.0683	0.0743
81	0.1241	0.1214	-0.0753	0.0720
82	0.1382	0.1287	-0.0827	0.0695
83	0.1538	0.1364	-0.0905	0.0667
84	0.1712	0.1445	-0.0986	0.0636
85	0.1906	0.1530	-0.1069	0.0602
86	0.2122	0.1620	-0.1154	0.0564
87	0.2363	0.1714	-0.1240	0.0523
88	0.2631	0.1812	-0.1328	0.0478
89	0.2929	0.1915	-0.1416	0.0430
90	0.3261	0.2021	-0.1505	0.0379
91	0.3630	0.2130	-0.1593	0.0325
92	0.4041	0.2241	-0.1680	0.0270
93	0.4498	0.2352	-0.1766	0.0213
94	0.5008	0.2464	-0.1849	0.0154
95	0.5574	0.2575	-0.1930	0.0095
96	0.6203	0.2684	-0.2009	0.0035
97	0.6899	0.2787	-0.2083	-0.0026
98	0.7667	0.2883	-0.2155	-0.0087
99	0.8513	0.2966	-0.2223	-0.0148
100	0.9443	0.3031	-0.2287	-0.0208
101	1.0463	0.3072	-0.2348	-0.0268
102	1.1579	0.3087	-0.2405	-0.0326
103	1.2794	0.3069	-0.2458	-0.0384
104	1.4109	0.3015	-0.2508	-0.0440
105	1.5521	0.2920	-0.2554	-0.0494
106	1.7027	0.2781	-0.2596	-0.0546
107	1.8617	0.2594	-0.2635	-0.0596
108	2.0283	0.2358	-0.2669	-0.0644
109	2.2015	0.2084	-0.2700	-0.0691
110	2.3783	0.1792	-0.2728	-0.0736
111	2.5547	0.1492	-0.2753	-0.0780
112	2.7268	0.1188	-0.2774	-0.0823
113	2.8912	0.0887	-0.2794	-0.0864
114	3.0446	0.0595	-0.2811	-0.0903
115	3.1856	0.0312	-0.2826	-0.0941
116	3.3167	0.0085	-0.2839	-0.0978
117	3.4271	0.0022	-0.2851	-0.1013
118	3.5060	0.0014	-0.2861	-0.1046

119	3.5594	0.0007	-0.2869	-0.1078
120	3.5901	0.0001	-0.2877	-0.1108
121	3.6000	0.0000	-0.2882	-0.1137

*** The Area of Airfoil = 0.97273811185982
*** The Area of Slat = 1.9050393977915E-2

```
     ---------------------------------------
     |                                     |
     |         OUTPUT FROM   ZONST         |
     |                                     |
     ---------------------------------------
```

```
NSING NA NS =  21,  29,  20
*** START ***

 ---------------------------------------
  ** NT =1  T = 2.E-3  ALP = 15.
 The Maximum Surface Vorticity Fourier Coeff.
           AIRFOIL                        SLAT
 ITER    AGA1_A    BGA1_A,        AGA1_S      BGA1_S
 1     -0.01174  -0.00227       -0.00462   0.00358
 2     -0.02085  -0.00422       -0.00720   0.00545
 3     -0.02791  -0.00590       -0.00862   0.00639
 4     -0.03338  -0.00734       -0.00941   0.00681
 5     -0.03763  -0.00858       -0.00984   0.00695
 6     -0.04094  -0.00965       -0.01007  -0.00722
 7     -0.04351  -0.01058       -0.01020  -0.00750
 8     -0.04552  -0.01137       -0.01026  -0.00773
 9     -0.04708  -0.01206       -0.01029  -0.00792
10     -0.04829  -0.01266       -0.01030  -0.00808
11     -0.04924  -0.01318       -0.01030  -0.00823
12     -0.04999  -0.01363       -0.01030  -0.00835
13     -0.05057  -0.01402       -0.01029  -0.00846
14     -0.05103  -0.01436       -0.01028  -0.00856
15     -0.05139  -0.01466       -0.01028  -0.00865
16     -0.05167  -0.01492       -0.01027  -0.00873
17     -0.05189  -0.01515       -0.01026  -0.00881
18     -0.05206  -0.01535       -0.01026  -0.00887
19     -0.05219  -0.01553       -0.01025  -0.00893
20     -0.05230  -0.01568       -0.01025  -0.00898
21     -0.05238  -0.01581       -0.01024  -0.00903
22     -0.05245  -0.01593       -0.01024  -0.00907
23     -0.05250  -0.01604       -0.01024  -0.00911
24     -0.05254  -0.01613       -0.01023  -0.00914
25     -0.05257  -0.01621       -0.01023  -0.00917
26     -0.05259  -0.01628       -0.01023  -0.00920
27     -0.05261  -0.01634       -0.01022  -0.00922
28     -0.05262  -0.01640       -0.01022  -0.00924
29     -0.05263  -0.01645       -0.01022  -0.00926
30     -0.05264  -0.01649       -0.01022  -0.00928
31     -0.05265  -0.01653       -0.01022  -0.00930
32     -0.05265  -0.01656       -0.01021  -0.00931
33     -0.05265  -0.01659       -0.01021  -0.00932
34     -0.05265  -0.01662       -0.01021  -0.00934
VORLS VORLA GAMAS = 3.7864931777192E-5,  -1.4041604562032E-14,  -3.7864647134285E-5

 ---------------------------------------
  ** NT =2  T = 4.E-3  ALP = 15.
 The Maximum Surface Vorticity Fourier Coeff.
           AIRFOIL                        SLAT
 ITER    AGA1_A    BGA1_A,        AGA1_S      BGA1_S
 1     -0.06028  -0.01824       -0.01189  -0.01053
 2     -0.06558  -0.01965       -0.01256  -0.01117
 3     -0.06931  -0.02087       -0.01288  -0.01161
 4     -0.07198  -0.02194       -0.01309  -0.01196
 5     -0.07394  -0.02286       -0.01324  -0.01226
 6     -0.07539  -0.02366       -0.01336  -0.01253
 7     -0.07648  -0.02436       -0.01347  -0.01276
 8     -0.07733  -0.02497       -0.01357  -0.01297
 9     -0.07799  -0.02550       -0.01366  -0.01316
10     -0.07852  -0.02597       -0.01374  -0.01333
```

11	-0.07895	-0.02638	-0.01381	-0.01349
12	-0.07930	-0.02673	-0.01387	-0.01362
13	-0.07959	-0.02705	-0.01393	-0.01375
14	-0.07982	-0.02733	-0.01398	-0.01386
15	-0.08003	-0.02757	-0.01402	-0.01396
16	-0.08020	-0.02779	-0.01406	-0.01405
17	-0.08034	-0.02798	-0.01410	-0.01413
18	-0.08047	-0.02814	-0.01413	-0.01420
19	-0.08057	-0.02829	-0.01416	-0.01427
20	-0.08066	-0.02842	-0.01419	-0.01432
21	-0.08074	-0.02854	-0.01421	-0.01438
22	-0.08081	-0.02864	-0.01424	-0.01442
23	-0.08087	-0.02873	-0.01426	-0.01447
24	-0.08092	-0.02881	-0.01427	-0.01450
25	-0.08097	-0.02888	-0.01429	-0.01454
26	-0.08101	-0.02894	-0.01430	-0.01457
27	-0.08104	-0.02899	-0.01431	-0.01460
28	-0.08108	-0.02904	-0.01433	-0.01462
29	-0.08110	-0.02909	-0.01434	-0.01464
30	-0.08113	-0.02912	-0.01435	-0.01466
31	-0.08115	-0.02916	-0.01435	-0.01468
32	-0.08117	-0.02919	-0.01436	-0.01470
33	-0.08118	-0.02922	-0.01437	-0.01471
34	-0.08120	-0.02924	-0.01437	-0.01473
35	-0.08121	-0.02926	-0.01438	-0.01474

VORLS VORLA GAMAS = 2.5480416613518E-4, -4.254904513923E-14, -1.7885802370621E-4

** NT =3 T = 6.E-3 ALP = 15.

The Maximum Surface Vorticity Fourier Coeff.

	AIRFOIL		SLAT	
ITER	AGA1_A	BGA1_A,	AGA1_S	BGA1_S
1	-0.08453	-0.03049	-0.01467	-0.01523
2	-0.08679	-0.03156	-0.01486	-0.01564
3	-0.08838	-0.03250	-0.01502	-0.01599
4	-0.08955	-0.03332	-0.01515	-0.01631
5	-0.09044	-0.03404	-0.01526	-0.01659
6	-0.09114	-0.03467	-0.01535	-0.01684
7	-0.09169	-0.03523	-0.01544	-0.01707
8	-0.09215	-0.03572	-0.01552	-0.01727
9	-0.09253	-0.03615	-0.01559	-0.01745
10	-0.09285	-0.03653	-0.01565	-0.01761
11	-0.09312	-0.03687	-0.01571	-0.01776
12	-0.09336	-0.03716	-0.01576	-0.01789
13	-0.09356	-0.03742	-0.01580	-0.01801
14	-0.09373	-0.03766	-0.01584	-0.01812
15	-0.09388	-0.03786	-0.01588	-0.01821
16	-0.09402	-0.03804	-0.01591	-0.01830
17	-0.09413	-0.03820	-0.01594	-0.01838
18	-0.09423	-0.03834	-0.01597	-0.01845
19	-0.09432	-0.03847	-0.01599	-0.01851
20	-0.09440	-0.03858	-0.01601	-0.01856
21	-0.09447	-0.03868	-0.01603	-0.01861
22	-0.09453	-0.03877	-0.01605	-0.01866
23	-0.09458	-0.03884	-0.01607	-0.01870
24	-0.09463	-0.03891	-0.01608	-0.01874
25	-0.09467	-0.03897	-0.01609	-0.01877
26	-0.09471	-0.03903	-0.01610	-0.01880
27	-0.09474	-0.03908	-0.01611	-0.01883
28	-0.09477	-0.03912	-0.01612	-0.01885
29	-0.09479	-0.03916	-0.01613	-0.01887
30	-0.09482	-0.03919	-0.01614	-0.01889
31	-0.09484	-0.03922	-0.01614	-0.01891
32	-0.09486	-0.03925	-0.01615	-0.01892
33	-0.09487	-0.03927	-0.01616	-0.01894
34	-0.09489	-0.03929	-0.01616	-0.01895

VORLS VORLA GAMAS = 8.683419575669E-4, -7.1200289934093E-14, -4.321829560817E-4

** NT =4 T = 8.E-3 ALP = 15.

The Maximum Surface Vorticity Fourier Coeff.

	AIRFOIL		SLAT	
ITER	AGA1_A	BGA1_A,	AGA1_S	BGA1_S
1	-0.09635	-0.04032	-0.01621	-0.01936
2	-0.09742	-0.04122	-0.01623	-0.01971
3	-0.09823	-0.04201	-0.01625	-0.02003
4	-0.09887	-0.04271	-0.01628	-0.02032
5	-0.09938	-0.04332	-0.01631	-0.02058
6	-0.09980	-0.04386	-0.01633	-0.02082
7	-0.10016	-0.04434	-0.01636	-0.02103
8	-0.10046	-0.04477	-0.01638	-0.02122
9	-0.10072	-0.04514	-0.01640	-0.02139
10	-0.10094	-0.04547	-0.01642	-0.02155
11	-0.10113	-0.04576	-0.01643	-0.02169
12	-0.10130	-0.04602	-0.01645	-0.02181
13	-0.10145	-0.04625	-0.01646	-0.02193
14	-0.10158	-0.04646	-0.01647	-0.02203
15	-0.10169	-0.04664	-0.01648	-0.02212
16	-0.10178	-0.04680	-0.01649	-0.02220
17	-0.10187	-0.04694	-0.01650	-0.02227
18	-0.10195	-0.04706	-0.01651	-0.02234
19	-0.10202	-0.04717	-0.01651	-0.02240
20	-0.10207	-0.04727	-0.01652	-0.02245
21	-0.10213	-0.04736	-0.01653	-0.02250
22	-0.10217	-0.04744	-0.01653	-0.02254
23	-0.10222	-0.04751	-0.01653	-0.02258
24	-0.10225	-0.04757	-0.01654	-0.02262
25	-0.10229	-0.04763	-0.01654	-0.02265
26	-0.10232	-0.04767	-0.01654	-0.02268
27	-0.10234	-0.04772	-0.01655	-0.02270
28	-0.10236	-0.04776	-0.01655	-0.02272
29	-0.10239	-0.04779	-0.01655	-0.02274
30	-0.10240	-0.04782	-0.01655	-0.02276
31	-0.10242	-0.04785	-0.01656	-0.02278
32	-0.10244	-0.04787	-0.01656	-0.02279
33	-0.10245	-0.04789	-0.01656	-0.02281

VORLS VORLA GAMAS = 2.0461174479292E-3, -7.2299962508457E-11, -7.3286621131342E-4

** NT =5 T = 1.E-2 ALP = 15.

The Maximum Surface Vorticity Fourier Coeff.

	AIRFOIL		SLAT	
ITER	AGA1_A	BGA1_A,	AGA1_S	BGA1_S
1	-0.10325	-0.04878	-0.01649	-0.02319
2	-0.10387	-0.04956	-0.01639	-0.02351
3	-0.10437	-0.05025	-0.01631	-0.02381
4	-0.10477	-0.05086	-0.01626	-0.02409
5	-0.10511	-0.05140	-0.01621	-0.02434
6	-0.10540	-0.05188	-0.01617	-0.02457
7	-0.10564	-0.05230	-0.01614	-0.02477
8	-0.10586	-0.05267	-0.01612	-0.02496
9	-0.10604	-0.05300	-0.01609	-0.02512
10	-0.10620	-0.05330	-0.01607	-0.02527
11	-0.10634	-0.05356	-0.01605	-0.02541
12	-0.10646	-0.05379	-0.01603	-0.02553
13	-0.10656	-0.05399	-0.01602	-0.02564
14	-0.10666	-0.05417	-0.01601	-0.02574
15	-0.10674	-0.05433	-0.01599	-0.02583
16	-0.10681	-0.05448	-0.01598	-0.02591
17	-0.10688	-0.05460	-0.01597	-0.02598
18	-0.10693	-0.05472	-0.01596	-0.02604
19	-0.10698	-0.05482	-0.01596	-0.02610

20	-0.10703	-0.05491	-0.01595	-0.02615
21	-0.10707	-0.05498	-0.01594	-0.02620
22	-0.10710	-0.05506	-0.01594	-0.02624
23	-0.10713	-0.05512	-0.01593	-0.02628
24	-0.10716	-0.05517	-0.01593	-0.02631
25	-0.10719	-0.05522	-0.01592	-0.02634
26	-0.10721	-0.05527	-0.01592	-0.02637
27	-0.10723	-0.05531	-0.01591	-0.02639
28	-0.10725	-0.05534	-0.01591	-0.02642
29	-0.10726	-0.05537	-0.01591	-0.02643
30	-0.10728	-0.05540	-0.01590	-0.02645
31	-0.10729	-0.05542	-0.01590	-0.02647

VORLS VORLA GAMAS = 3.8132464603259E-3, -7.1449607387208E-10, -9.9187462373854E-4

** NT =6 T = 1.2E-2 ALP = 15.

The Maximum Surface Vorticity Fourier Coeff.

	AIRFOIL		SLAT	
ITER	AGA1_A	BGA1_A,	AGA1_S	BGA1_S
1	-0.10781	-0.05622	-0.01576	-0.02685
2	-0.10823	-0.05692	-0.01557	-0.02715
3	-0.10858	-0.05755	-0.01543	-0.02744
4	-0.10887	-0.05810	-0.01532	-0.02771
5	-0.10912	-0.05858	-0.01522	-0.02795
6	-0.10933	-0.05902	-0.01514	-0.02817
7	-0.10951	-0.05940	-0.01507	-0.02838
8	-0.10966	-0.05974	-0.01501	-0.02856
9	-0.10980	-0.06004	-0.01496	-0.02872
10	-0.10992	-0.06031	-0.01491	-0.02887
11	-0.11002	-0.06054	-0.01487	-0.02901
12	-0.11012	-0.06075	-0.01483	-0.02913
13	-0.11020	-0.06094	-0.01479	-0.02923
14	-0.11027	-0.06110	-0.01476	-0.02933
15	-0.11033	-0.06125	-0.01474	-0.02942
16	-0.11039	-0.06138	-0.01471	-0.02950
17	-0.11044	-0.06150	-0.01469	-0.02957
18	-0.11048	-0.06160	-0.01467	-0.02963
19	-0.11052	-0.06169	-0.01465	-0.02969
20	-0.11055	-0.06178	-0.01463	-0.02974
21	-0.11059	-0.06185	-0.01462	-0.02978
22	-0.11061	-0.06191	-0.01460	-0.02983
23	-0.11064	-0.06197	-0.01459	-0.02986
24	-0.11066	-0.06202	-0.01458	-0.02990
25	-0.11068	-0.06207	-0.01457	-0.02993
26	-0.11070	-0.06211	-0.01456	-0.02995
27	-0.11071	-0.06214	-0.01456	-0.02998
28	-0.11073	-0.06218	-0.01455	-0.03000
29	-0.11074	-0.06220	-0.01454	-0.03002
30	-0.11075	-0.06223	-0.01454	-0.03004

VORLS VORLA GAMAS = 6.0483802073049E-3, -4.9244655368289E-9, -1.144225878819E-3

** NT =7 T = 1.4E-2 ALP = 15.

The Maximum Surface Vorticity Fourier Coeff.

	AIRFOIL		SLAT	
ITER	AGA1_A	BGA1_A,	AGA1_S	BGA1_S
1	-0.11113	-0.06295	-0.01437	-0.03041
2	-0.11144	-0.06358	-0.01413	-0.03070
3	-0.11170	-0.06415	-0.01395	-0.03099
4	-0.11192	-0.06465	-0.01381	-0.03125
5	-0.11210	-0.06509	-0.01369	-0.03149
6	-0.11227	-0.06548	-0.01359	-0.03171
7	-0.11241	-0.06583	-0.01350	-0.03191
8	-0.11253	-0.06614	-0.01342	-0.03209
9	-0.11264	-0.06641	-0.01335	-0.03225
10	-0.11274	-0.06665	-0.01329	-0.03240

11	-0.11282	-0.06687	-0.01323	-0.03253
12	-0.11289	-0.06706	0.01324	-0.03265
13	-0.11296	-0.06723	0.01327	-0.03276
14	-0.11302	-0.06738	0.01329	-0.03286
15	-0.11307	-0.06752	0.01331	-0.03294
16	-0.11311	-0.06764	0.01333	-0.03302
17	-0.11315	-0.06774	0.01335	-0.03309
18	-0.11319	-0.06784	0.01336	-0.03316
19	-0.11322	-0.06792	0.01338	-0.03321
20	-0.11325	-0.06800	0.01339	-0.03326
21	-0.11328	-0.06806	0.01340	-0.03331
22	-0.11330	-0.06812	0.01341	-0.03335
23	-0.11332	-0.06818	0.01342	-0.03339
24	-0.11334	-0.06822	0.01343	-0.03342
25	-0.11335	-0.06827	0.01344	-0.03345
26	-0.11337	-0.06830	0.01344	-0.03348
27	-0.11338	-0.06834	0.01345	-0.03350
28	-0.11339	-0.06837	0.01346	-0.03352
29	-0.11340	-0.06839	0.01346	-0.03354

VORLS VORLA GAMAS = 8.5503227680112E-3, -2.9133514424554E-8, -1.1699102569524E-3

** NT =8 T = 1.6E-2 ALP = 15.

The Maximum Surface Vorticity Fourier Coeff.

	AIRFOIL		SLAT	
ITER	AGA1_A	BGA1_A,	AGA1_S	BGA1_S
1	-0.11369	-0.06904	0.01353	-0.03392
2	-0.11394	-0.06962	0.01356	-0.03421
3	-0.11415	-0.07013	0.01360	-0.03448
4	-0.11433	-0.07058	0.01365	-0.03475
5	-0.11448	-0.07098	0.01369	-0.03499
6	-0.11462	-0.07134	0.01373	-0.03521
7	-0.11474	-0.07165	0.01377	-0.03541
8	-0.11484	-0.07193	0.01381	-0.03559
9	-0.11493	-0.07218	0.01384	-0.03575
10	-0.11501	-0.07240	0.01387	-0.03590
11	-0.11508	-0.07260	0.01389	-0.03603
12	-0.11515	-0.07278	0.01392	-0.03615
13	-0.11520	-0.07293	0.01394	-0.03626
14	-0.11525	-0.07307	0.01396	-0.03635
15	-0.11530	-0.07319	0.01398	-0.03644
16	-0.11534	-0.07330	0.01399	-0.03652
17	-0.11537	-0.07340	0.01401	-0.03659
18	-0.11540	-0.07349	0.01402	-0.03665
19	-0.11543	-0.07356	0.01403	-0.03671
20	-0.11546	-0.07363	0.01404	-0.03676
21	-0.11548	-0.07369	0.01405	-0.03681
22	-0.11550	-0.07375	0.01406	-0.03685
23	-0.11552	-0.07380	0.01407	-0.03689
24	-0.11553	-0.07384	0.01407	-0.03692
25	-0.11555	-0.07388	0.01408	-0.03695
26	-0.11556	-0.07391	0.01409	-0.03698
27	-0.11557	-0.07395	0.01409	-0.03700
28	-0.11558	-0.07397	0.01410	-0.03702

VORLS VORLA GAMAS = 1.1129904112676E-2, -1.3437515284584E-7, -1.077353289232E-3

** NT =9 T = 1.8E-2 ALP = 15.

The Maximum Surface Vorticity Fourier Coeff.

	AIRFOIL		SLAT	
ITER	AGA1_A	BGA1_A,	AGA1_S	BGA1_S
1	-0.11582	-0.07456	0.01415	-0.03740
2	-0.11603	-0.07509	0.01416	-0.03770
3	-0.11620	-0.07556	0.01418	-0.03798
4	-0.11636	-0.07597	0.01421	-0.03824
5	-0.11649	-0.07634	0.01424	-0.03849

6	-0.11661	-0.07666	0.01427	-0.03871
7	-0.11671	-0.07695	0.01430	-0.03891
8	-0.11680	-0.07721	0.01433	-0.03910
9	-0.11688	-0.07744	0.01435	-0.03926
10	-0.11695	-0.07765	0.01437	-0.03941
11	-0.11701	-0.07783	0.01439	-0.03955
12	-0.11707	-0.07799	0.01441	-0.03967
13	-0.11712	-0.07813	0.01443	-0.03978
14	-0.11716	-0.07826	0.01444	-0.03987
15	-0.11720	-0.07837	0.01446	-0.03996
16	-0.11724	-0.07847	0.01447	-0.04004
17	-0.11727	-0.07856	0.01448	-0.04011
18	-0.11730	-0.07865	0.01449	-0.04018
19	-0.11732	-0.07872	0.01450	-0.04023
20	-0.11735	-0.07878	0.01450	-0.04029
21	-0.11737	-0.07884	0.01451	-0.04033
22	-0.11738	-0.07889	0.01452	-0.04037
23	-0.11740	-0.07893	0.01452	-0.04041
24	-0.11741	-0.07897	0.01453	-0.04045
25	-0.11743	-0.07901	0.01453	-0.04048
26	-0.11744	-0.07904	0.01454	-0.04050
27	-0.11745	-0.07907	0.01454	-0.04053
28	-0.11746	-0.07910	0.01454	-0.04055

VORLS VORLA GAMAS = 1.3633716551865E-2, -4.9995902940263E-7, -9.3334397687046E-4

** NT =10 T = 2.E-2 ALP = 15.

The Maximum Surface Vorticity Fourier Coeff.

	AIRFOIL		SLAT	
ITER	AGA1_A	BGA1_A,	AGA1_S	BGA1_S
1	-0.11766	-0.07964	0.01458	-0.04093
2	-0.11784	-0.08012	0.01457	-0.04124
3	-0.11799	-0.08055	0.01458	-0.04152
4	-0.11812	-0.08092	0.01460	-0.04179
5	-0.11826	-0.08126	0.01462	-0.04204
6	-0.11853	-0.08156	0.01464	-0.04226
7	-0.11878	-0.08183	0.01465	-0.04247
8	-0.11899	-0.08206	0.01467	-0.04265
9	-0.11919	-0.08228	0.01469	-0.04282
10	-0.11936	-0.08246	0.01470	-0.04297
11	-0.11951	-0.08263	0.01472	-0.04311
12	-0.11965	-0.08278	0.01473	-0.04323
13	-0.11978	-0.08291	0.01474	-0.04334
14	-0.11989	-0.08303	0.01475	-0.04344
15	-0.11998	-0.08313	0.01476	-0.04353
16	-0.12007	-0.08323	0.01476	-0.04361
17	-0.12015	-0.08331	0.01477	-0.04368
18	-0.12022	-0.08339	0.01478	-0.04374
19	-0.12028	-0.08345	0.01478	-0.04380
20	-0.12034	-0.08351	0.01479	-0.04385
21	-0.12039	-0.08356	0.01479	-0.04390
22	-0.12043	-0.08361	0.01480	-0.04394
23	-0.12047	-0.08365	0.01480	-0.04398
24	-0.12051	-0.08369	0.01481	-0.04402
25	-0.12054	-0.08372	0.01481	-0.04405
26	-0.12057	-0.08375	0.01481	-0.04407
27	-0.12059	-0.08378	0.01481	-0.04410

VORLS VORLA GAMAS = 1.5979317059393E-2, -1.7367179190689E-6, -7.9290859594667E-4

** NT =11 T = 2.2E-2 ALP = 15.

The Maximum Surface Vorticity Fourier Coeff.

	AIRFOIL		SLAT	
ITER	AGA1_A	BGA1_A,	AGA1_S	BGA1_S
1	-0.12105	-0.08429	0.01483	-0.04449
2	-0.12145	-0.08474	0.01481	-0.04480

3	-0.12181	-0.08514	0.01481	-0.04510
4	-0.12213	-0.08549	0.01481	-0.04537
5	-0.12241	-0.08581	0.01481	-0.04562
6	-0.12267	-0.08609	0.01482	-0.04586
7	-0.12289	-0.08634	0.01483	-0.04606
8	-0.12309	-0.08657	0.01484	-0.04625
9	-0.12327	-0.08676	0.01485	-0.04642
10	-0.12343	-0.08694	0.01485	-0.04658
11	-0.12358	-0.08710	0.01486	-0.04672
12	-0.12370	-0.08724	0.01487	-0.04684
13	-0.12382	-0.08736	0.01487	-0.04695
14	-0.12392	-0.08747	0.01488	-0.04705
15	-0.12401	-0.08757	0.01488	-0.04715
16	-0.12409	-0.08766	0.01489	-0.04723
17	-0.12417	-0.08774	0.01489	-0.04730
18	-0.12423	-0.08781	0.01489	-0.04737
19	-0.12429	-0.08787	0.01490	-0.04743
20	-0.12434	-0.08793	0.01490	-0.04748
21	-0.12439	-0.08798	0.01490	-0.04753
22	-0.12443	-0.08802	0.01490	-0.04757
23	-0.12447	-0.08806	0.01491	-0.04761
24	-0.12450	-0.08810	0.01491	-0.04764
25	-0.12453	-0.08813	0.01491	-0.04767
26	-0.12456	-0.08816	0.01491	-0.04770

VORLS VORLA GAMAS = 1.8126815994758E-2, -4.8320368310957E-6, -6.4500012526942E-4

** NT =12 T = 2.4E-2 ALP = 15.

The Maximum Surface Vorticity Fourier Coeff.

	AIRFOIL		SLAT	
ITER	AGA1_A	BGA1_A,	AGA1_S	BGA1_S
1	-0.12498	-0.08864	0.01491	-0.04811
2	-0.12536	-0.08907	0.01488	-0.04843
3	-0.12569	-0.08945	0.01486	-0.04873
4	-0.12599	-0.08978	0.01485	-0.04902
5	-0.12626	-0.09008	0.01485	-0.04928
6	-0.12650	-0.09035	0.01484	-0.04951
7	-0.12671	-0.09059	0.01484	-0.04973
8	-0.12690	-0.09080	0.01484	-0.04992
9	-0.12707	-0.09099	0.01484	-0.05010
10	-0.12722	-0.09115	0.01484	-0.05025
11	-0.12736	-0.09130	0.01484	-0.05040
12	-0.12748	-0.09144	0.01484	-0.05052
13	-0.12758	-0.09156	0.01484	-0.05064
14	-0.12768	-0.09166	0.01484	-0.05074
15	-0.12777	-0.09176	0.01485	-0.05083
16	-0.12784	-0.09184	0.01485	-0.05092
17	-0.12791	-0.09192	0.01485	-0.05099
18	-0.12797	-0.09198	0.01485	-0.05106
19	-0.12803	-0.09204	0.01485	-0.05112
20	-0.12808	-0.09210	0.01485	-0.05118
21	-0.12812	-0.09215	0.01485	-0.05122
22	-0.12816	-0.09219	0.01485	-0.05127
23	-0.12819	-0.09223	0.01485	-0.05131
24	-0.12823	-0.09226	0.01485	-0.05134
25	-0.12825	-0.09229	0.01485	-0.05138
26	-0.12828	-0.09232	0.01485	-0.05140

VORLS VORLA GAMAS = 2.0080855747628E-2, -1.1246599984394E-5, -5.0167075009813E-4

** NT =13 T = 2.6E-2 ALP = 15.

The Maximum Surface Vorticity Fourier Coeff.

	AIRFOIL		SLAT	
ITER	AGA1_A	BGA1_A,	AGA1_S	BGA1_S
1	-0.12867	-0.09276	0.01483	-0.05182
2	-0.12902	-0.09316	0.01479	-0.05215

3	-0.12933	-0.09351	0.01476	-0.05246
4	-0.12961	-0.09383	0.01474	-0.05275
5	-0.12986	-0.09411	0.01472	-0.05302
6	-0.13008	-0.09436	0.01471	-0.05326
7	-0.13028	-0.09458	0.01470	-0.05348
8	-0.13045	-0.09478	0.01469	-0.05367
9	-0.13061	-0.09495	0.01469	-0.05385
10	-0.13075	-0.09511	0.01468	-0.05401
11	-0.13088	-0.09525	0.01467	-0.05415
12	-0.13099	-0.09538	0.01467	-0.05428
13	-0.13109	-0.09549	0.01467	-0.05440
14	-0.13118	-0.09559	0.01466	-0.05451
15	-0.13126	-0.09568	0.01466	-0.05460
16	-0.13133	-0.09576	0.01466	-0.05468
17	-0.13139	-0.09583	0.01465	-0.05476
18	-0.13145	-0.09589	0.01465	-0.05483
19	-0.13150	-0.09595	0.01465	-0.05489
20	-0.13155	-0.09600	0.01465	-0.05495
21	-0.13159	-0.09604	0.01465	-0.05500
22	-0.13163	-0.09608	0.01465	-0.05504
23	-0.13166	-0.09612	0.01464	-0.05508
24	-0.13169	-0.09615	0.01464	-0.05512
25	-0.13171	-0.09618	0.01464	-0.05515
26	-0.13174	-0.09620	0.01464	-0.05518

VORLS VORLA GAMAS = 2.18603542087E-2, -2.3342652173493E-5, -3.8985833785652E-4

** NT =14 T = 2.8E-2 ALP = 15.

The Maximum Surface Vorticity Fourier Coeff.

	AIRFOIL		SLAT	
ITER	AGA1_A	BGA1_A,	AGA1_S	BGA1_S
1	-0.13210	-0.09662	0.01461	-0.05560
2	-0.13243	-0.09699	0.01455	-0.05594
3	-0.13272	-0.09732	0.01451	-0.05626
4	-0.13298	-0.09762	0.01448	-0.05656
5	-0.13321	-0.09788	0.01446	-0.05683
6	-0.13342	-0.09811	0.01444	-0.05707
7	-0.13360	-0.09832	0.01442	-0.05729
8	-0.13376	-0.09850	0.01440	-0.05749
9	-0.13391	-0.09867	0.01439	-0.05767
10	-0.13404	-0.09882	0.01438	-0.05783
11	-0.13416	-0.09895	0.01437	-0.05798
12	-0.13427	-0.09907	0.01436	-0.05811
13	-0.13436	-0.09917	0.01435	-0.05823
14	-0.13444	-0.09926	0.01434	-0.05834
15	-0.13452	-0.09935	0.01434	-0.05843
16	-0.13459	-0.09942	0.01433	-0.05852
17	-0.13465	-0.09949	0.01433	-0.05860
18	-0.13470	-0.09955	0.01432	-0.05867
19	-0.13475	-0.09960	0.01432	-0.05873
20	-0.13479	-0.09965	0.01431	-0.05878
21	-0.13483	-0.09969	0.01431	-0.05883
22	-0.13486	-0.09973	0.01431	-0.05888
23	-0.13489	-0.09976	0.01430	-0.05892
24	-0.13492	-0.09979	0.01430	-0.05896
25	-0.13495	-0.09982	0.01430	-0.05899
26	-0.13497	-0.09984	0.01430	-0.05902

VORLS VORLA GAMAS = 2.3487787663998E-2, -4.4369433464326E-5, -2.8361843447879E-4

** NT =15 T = 3.E-2 ALP = 15.

The Maximum Surface Vorticity Fourier Coeff.

	AIRFOIL		SLAT	
ITER	AGA1_A	BGA1_A,	AGA1_S	BGA1_S
1	-0.13531	-0.10024	0.01425	-0.05944
2	-0.13562	-0.10059	0.01419	-0.05980

3	-0.13590	-0.10091	0.01414	-0.06013
4	-0.13614	-0.10119	0.01410	-0.06043
5	-0.13636	-0.10144	0.01406	-0.06070
6	-0.13656	-0.10166	0.01403	-0.06095
7	-0.13673	-0.10186	0.01401	-0.06118
8	-0.13689	-0.10203	0.01399	-0.06138
9	-0.13703	-0.10219	0.01397	-0.06156
10	-0.13715	-0.10233	0.01395	-0.06173
11	-0.13726	-0.10246	0.01393	-0.06188
12	-0.13736	-0.10257	0.01392	-0.06201
13	-0.13745	-0.10267	0.01391	-0.06213
14	-0.13753	-0.10276	0.01390	-0.06224
15	-0.13760	-0.10284	0.01389	-0.06233
16	-0.13767	-0.10291	0.01388	-0.06242
17	-0.13773	-0.10297	0.01387	-0.06250
18	-0.13778	-0.10303	0.01387	-0.06257
19	-0.13782	-0.10308	0.01386	-0.06263
20	-0.13786	-0.10313	0.01385	-0.06269
21	-0.13790	-0.10317	0.01385	-0.06274
22	-0.13793	-0.10320	0.01384	-0.06279
23	-0.13796	-0.10324	0.01384	-0.06283
24	-0.13799	-0.10327	0.01384	-0.06287
25	-0.13801	-0.10329	0.01383	-0.06290
26	-0.13803	-0.10331	0.01383	-0.06293
27	-0.13805	-0.10333	0.01383	-0.06296

VORLS VORLA GAMAS = 2.4992591083464E-2, -7.6094544859073E-5, -1.6294116727476E-4

** NT =16 T = 3.2E-2 ALP = 15.

The Maximum Surface Vorticity Fourier Coeff.

	AIRFOIL		SLAT	
ITER	AGA1_A	BGA1_A,	AGA1_S	BGA1_S
1	-0.13838	-0.10371	0.01377	-0.06338
2	-0.13867	-0.10405	0.01370	-0.06374
3	-0.13893	-0.10434	0.01364	-0.06407
4	-0.13916	-0.10461	0.01359	-0.06438
5	-0.13937	-0.10485	0.01355	-0.06466
6	-0.13955	-0.10506	0.01351	-0.06491
7	-0.13972	-0.10525	0.01348	-0.06513
8	-0.13986	-0.10541	0.01345	-0.06534
9	-0.14000	-0.10556	0.01343	-0.06552
10	-0.14011	-0.10570	0.01341	-0.06569
11	-0.14022	-0.10582	0.01339	-0.06584
12	-0.14031	-0.10592	0.01337	-0.06597
13	-0.14040	-0.10602	0.01336	-0.06609
14	-0.14047	-0.10610	0.01334	-0.06620
15	-0.14054	-0.10618	0.01333	-0.06630
16	-0.14060	-0.10625	0.01332	-0.06639
17	-0.14066	-0.10631	0.01331	-0.06647
18	-0.14070	-0.10636	0.01330	-0.06654
19	-0.14075	-0.10641	0.01329	-0.06660
20	-0.14079	-0.10645	0.01329	-0.06666
21	-0.14082	-0.10649	0.01328	-0.06671
22	-0.14085	-0.10653	0.01328	-0.06676
23	-0.14088	-0.10656	0.01327	-0.06680
24	-0.14090	-0.10659	0.01327	-0.06683
25	-0.14093	-0.10661	0.01326	-0.06687
26	-0.14095	-0.10663	0.01326	-0.06690
27	-0.14096	-0.10665	0.01325	-0.06692

VORLS VORLA GAMAS = 2.6398967789493E-2, -1.2102493351635E-4, -1.9216270863195E-5

** NT =17 T = 3.4E-2 ALP = 15.

The Maximum Surface Vorticity Fourier Coeff.

	AIRFOIL		SLAT	
ITER	AGA1_A	BGA1_A,	AGA1_S	BGA1_S

1	-0.14127	-0.10701	0.01318	-0.06736
2	-0.14154	-0.10732	0.01310	-0.06773
3	-0.14179	-0.10760	0.01303	-0.06806
4	-0.14201	-0.10785	0.01298	-0.06837
5	-0.14220	-0.10808	0.01293	-0.06865
6	-0.14238	-0.10828	0.01289	-0.06891
7	-0.14253	-0.10845	0.01285	-0.06914
8	-0.14267	-0.10861	0.01282	-0.06935
9	-0.14280	-0.10876	0.01279	-0.06953
10	-0.14291	-0.10888	0.01276	-0.06970
11	-0.14301	-0.10900	0.01274	-0.06985
12	-0.14310	-0.10910	0.01272	-0.06999
13	-0.14318	-0.10919	0.01270	-0.07011
14	-0.14325	-0.10927	0.01268	-0.07022
15	-0.14331	-0.10934	0.01267	-0.07032
16	-0.14337	-0.10940	0.01266	-0.07041
17	-0.14342	-0.10946	0.01264	-0.07049
18	-0.14346	-0.10951	0.01263	-0.07056
19	-0.14350	-0.10956	0.01262	-0.07062
20	-0.14354	-0.10960	0.01261	-0.07068
21	-0.14357	-0.10964	0.01261	-0.07073
22	-0.14360	-0.10967	0.01260	-0.07078
23	-0.14363	-0.10970	0.01259	-0.07082
24	-0.14365	-0.10972	0.01259	-0.07086
25	-0.14367	-0.10975	0.01258	-0.07089
26	-0.14369	-0.10977	0.01258	-0.07092
27	-0.14371	-0.10979	0.01258	-0.07095

VORLS VORLA GAMAS = 2.77266238012E-2, -1.8099409778814E-4, 1.0757527868319E-4

** NT =18 T = 3.6E-2 ALP = 15.

The Maximum Surface Vorticity Fourier Coeff.

	AIRFOIL		SLAT	
ITER	AGA1_A	BGA1_A,	AGA1_S	BGA1_S
1	-0.14400	-0.11013	0.01249	-0.07139
2	-0.14426	-0.11043	-0.01252	-0.07176
3	-0.14449	-0.11069	-0.01264	-0.07211
4	-0.14470	-0.11093	-0.01276	-0.07242
5	-0.14488	-0.11115	-0.01286	-0.07271
6	-0.14505	-0.11134	-0.01296	-0.07296
7	-0.14520	-0.11151	-0.01305	-0.07320
8	-0.14533	-0.11166	-0.01313	-0.07341
9	-0.14545	-0.11180	-0.01320	-0.07359
10	-0.14555	-0.11192	-0.01326	-0.07376
11	-0.14565	-0.11203	-0.01332	-0.07392
12	-0.14573	-0.11212	-0.01337	-0.07405
13	-0.14581	-0.11221	-0.01342	-0.07418
14	-0.14588	-0.11229	-0.01346	-0.07429
15	-0.14594	-0.11236	-0.01350	-0.07439
16	-0.14599	-0.11242	-0.01354	-0.07448
17	-0.14604	-0.11247	-0.01357	-0.07456
18	-0.14608	-0.11252	-0.01360	-0.07463
19	-0.14612	-0.11257	-0.01362	-0.07470
20	-0.14616	-0.11261	-0.01364	-0.07476
21	-0.14619	-0.11264	-0.01366	-0.07481
22	-0.14622	-0.11267	-0.01368	-0.07486
23	-0.14624	-0.11270	-0.01370	-0.07490
24	-0.14626	-0.11273	-0.01371	-0.07494
25	-0.14628	-0.11275	-0.01373	-0.07497
26	-0.14630	-0.11277	-0.01374	-0.07500
27	-0.14632	-0.11279	-0.01375	-0.07503

VORLS VORLA GAMAS = 2.8988467520735E-2, -2.5373716022074E-4, 1.8462938760463E-4

** NT =19 T = 3.8E-2 ALP = 15.

The Maximum Surface Vorticity Fourier Coeff.

	AIRFOIL		SLAT	
ITER	AGA1_A	BGA1_A,	AGA1_S	BGA1_S
1	-0.14659	-0.11311	-0.01392	-0.07547
2	-0.14683	-0.11339	-0.01406	-0.07585
3	-0.14705	-0.11364	-0.01419	-0.07620
4	-0.14725	-0.11387	-0.01431	-0.07652
5	-0.14742	-0.11407	-0.01442	-0.07681
6	-0.14758	-0.11425	-0.01452	-0.07707
7	-0.14772	-0.11441	-0.01462	-0.07730
8	-0.14784	-0.11456	-0.01470	-0.07751
9	-0.14795	-0.11469	-0.01477	-0.07771
10	-0.14805	-0.11480	-0.01484	-0.07788
11	-0.14814	-0.11490	-0.01490	-0.07803
12	-0.14822	-0.11500	-0.01496	-0.07817
13	-0.14829	-0.11508	-0.01501	-0.07830
14	-0.14836	-0.11515	-0.01505	-0.07841
15	-0.14841	-0.11522	-0.01509	-0.07851
16	-0.14847	-0.11528	-0.01513	-0.07860
17	-0.14851	-0.11533	-0.01516	-0.07868
18	-0.14855	-0.11537	-0.01519	-0.07875
19	-0.14859	-0.11542	-0.01521	-0.07882
20	-0.14862	-0.11545	-0.01524	-0.07888
21	-0.14865	-0.11549	-0.01526	-0.07893
22	-0.14868	-0.11552	-0.01528	-0.07898
23	-0.14870	-0.11554	-0.01529	-0.07902
24	-0.14872	-0.11557	-0.01531	-0.07906
25	-0.14874	-0.11559	-0.01532	-0.07910
26	-0.14876	-0.11561	-0.01534	-0.07913
27	-0.14877	-0.11563	-0.01535	-0.07915
28	-0.14879	-0.11564	-0.01536	-0.07918

VORLS VORLA GAMAS = 3.0200306750889E-2, -3.4964133060229E-4, 2.7581700119138E-4

** NT =20 T = 4.E-2 ALP = 15.

The Maximum Surface Vorticity Fourier Coeff.

	AIRFOIL		SLAT	
ITER	AGA1_A	BGA1_A,	AGA1_S	BGA1_S
1	-0.14905	-0.11595	-0.01553	-0.07963
2	-0.14928	-0.11622	-0.01568	-0.08001
3	-0.14948	-0.11646	-0.01582	-0.08035
4	-0.14967	-0.11668	-0.01594	-0.08067
5	-0.14983	-0.11687	-0.01606	-0.08096
6	-0.14998	-0.11704	-0.01616	-0.08122
7	-0.15011	-0.11720	-0.01626	-0.08146
8	-0.15023	-0.11733	-0.01634	-0.08167
9	-0.15034	-0.11746	-0.01642	-0.08186
10	-0.15043	-0.11757	-0.01649	-0.08204
11	-0.15051	-0.11766	-0.01655	-0.08219
12	-0.15059	-0.11775	-0.01661	-0.08233
13	-0.15066	-0.11783	-0.01666	-0.08245
14	-0.15072	-0.11790	-0.01671	-0.08257
15	-0.15077	-0.11796	-0.01675	-0.08267
16	-0.15082	-0.11802	-0.01678	-0.08276
17	-0.15086	-0.11807	-0.01682	-0.08284
18	-0.15090	-0.11811	-0.01685	-0.08291
19	-0.15094	-0.11815	-0.01687	-0.08298
20	-0.15097	-0.11819	-0.01690	-0.08304
21	-0.15100	-0.11822	-0.01692	-0.08309
22	-0.15102	-0.11825	-0.01694	-0.08314
23	-0.15105	-0.11828	-0.01696	-0.08318
24	-0.15107	-0.11830	-0.01697	-0.08322
25	-0.15108	-0.11832	-0.01699	-0.08325
26	-0.15110	-0.11834	-0.01700	-0.08328
27	-0.15111	-0.11835	-0.01701	-0.08331
28	-0.15113	-0.11837	-0.01702	-0.08334

VORLS VORLA GAMAS = 3.1370880402609E-2, -4.701584 B.24 5E-4, 3.8752551128173E-4

VORTICITY ---SLAT RING---CCW FROM TRAILING EDGE:

-214.79	-205.34	-194.43	-181.18	-166.59	-150.13	-132.07	-113.40
-93.16	-72.81	-50.70	-27.88	-2.10	25.95	59.36	97.70
144.14	200.50	268.41	355.58	469.61	620.80	824.82	1088.43
1385.65	1655.27	1847.25	1949.86	1950.09	1820.04	1545.53	1165.92
795.77	532.25	381.81	304.87	263.71	233.16	207.42	190.88
183.35	191.19	211.12	231.39	248.37	263.68	279.82	299.04
321.44	345.53	375.11	416.97	461.59	528.15	599.45	681.65
770.95	857.81	943.17	1025.88	1123.16	1237.14	1350.44	1494.92
1688.12	1950.43	2299.98	2787.46	4094.40	6567.91	60530.14	-2691.32
-1799.62	-1366.94	-1095.56	-921.48	-802.11	-716.35	-645.81	-581.96
-536.03	-502.31	-474.77	-452.59	-432.41	-411.89	-393.07	-375.92
-360.36	-346.76	-335.39	-326.35	-319.61	-314.88	-311.52	-308.70
-305.73	-302.16	-297.89	-293.41	-289.62	-286.73	-284.53	-283.83
-284.82	-286.35	-288.56	-289.85	-289.57	-287.36	-282.97	-277.28
-270.61	-263.69	-257.07	-250.27	-244.14	-237.43	-230.59	-223.37

VORTICITY ---AIRFOIL RING---CCW FROM TRAILING EDGE:

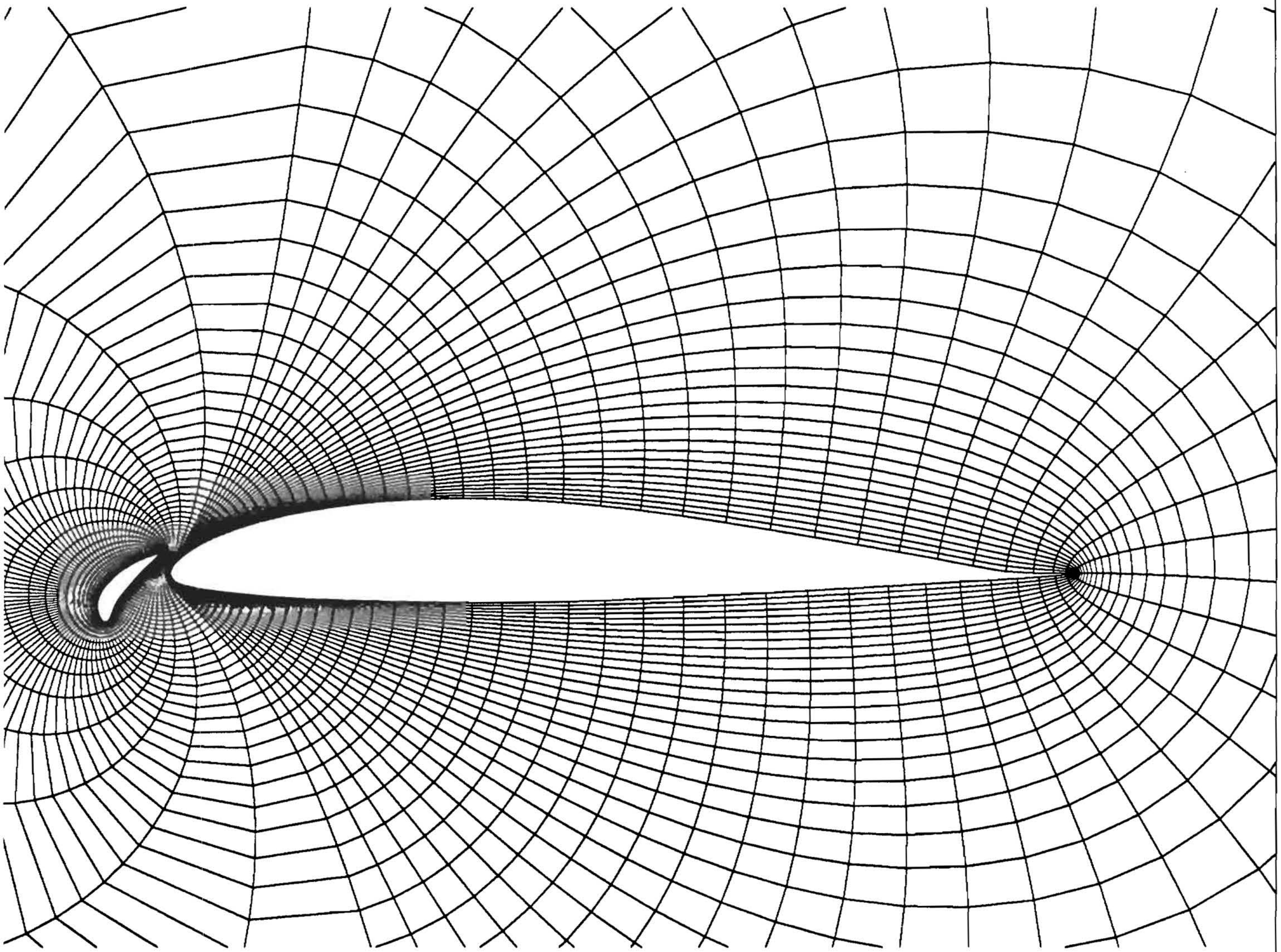
238.68	325.75	180.50	131.16	100.48	80.83	67.97	59.51
54.86	51.58	49.96	49.61	50.28	51.70	54.16	57.72
62.63	67.94	74.01	80.79	87.82	96.01	106.05	117.41
129.75	143.60	159.66	178.17	198.63	218.99	235.32	243.92
245.32	248.36	265.97	293.76	302.11	287.04	307.52	283.88
292.92	237.73	222.81	217.60	183.41	97.36	-4.72	-117.24
-254.69	-291.18	-381.57	-938.91	-775.86	-1473.97	-1270.04	-1088.50
-955.58	-763.54	-616.28	-571.17	-415.13	-263.50	-592.84	-306.76
-63.89	-137.90	-293.20	-434.80	-226.61	-74.64	-74.56	-244.23
-222.25	-169.53	-220.45	-326.91	-495.11	-621.09	-612.84	-610.29
-704.33	-648.42	-689.76	-627.94	-656.50	-632.64	-566.48	-534.04
-533.19	-530.83	-507.66	-464.51	-411.48	-357.96	-309.45	-267.83
-232.53	-201.73	-174.12	-149.59	-128.51	-110.56	-94.99	-81.87
-70.85	-61.38	-53.37	-46.33	-40.40	-36.03	-32.72	-30.06
-27.81	-25.71	-22.70	-19.81	-22.92	-36.84	-88.36	-249.92

```
--------------------------------------
|                                    |
|        OUTPUT FROM  LOADS          |
|                                    |
--------------------------------------
```

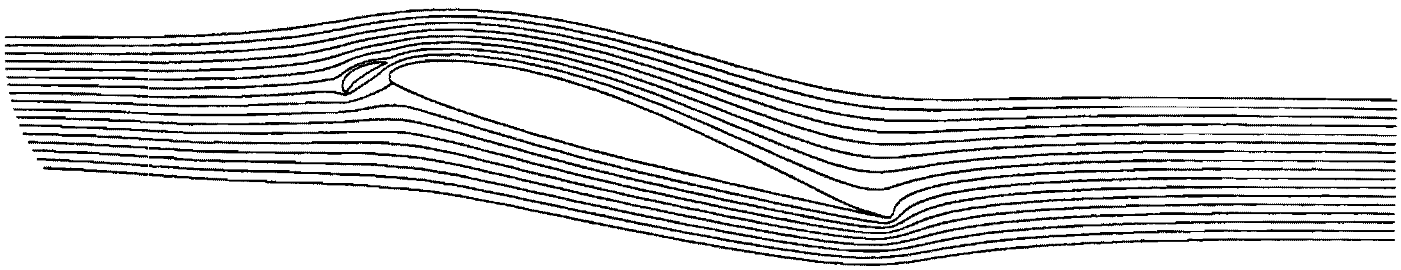
```
*** NT = 20   T = 4.E-2   ALP = 15.

*** Airfoil ***
CL CD CM = 0.6591823682288,  9.633987160583E-2,  -0.1007629885579

*** Slat ***
Cl CD CM = 6.4596823288944E-2,  -2.6228068260993E-2,  -1.4524851531978E-2
```

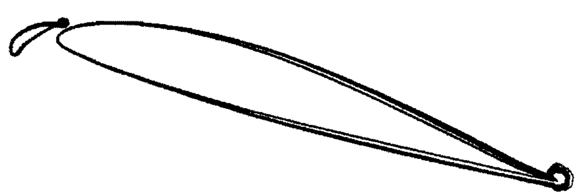
B.6 Sample Plots

From Plot1

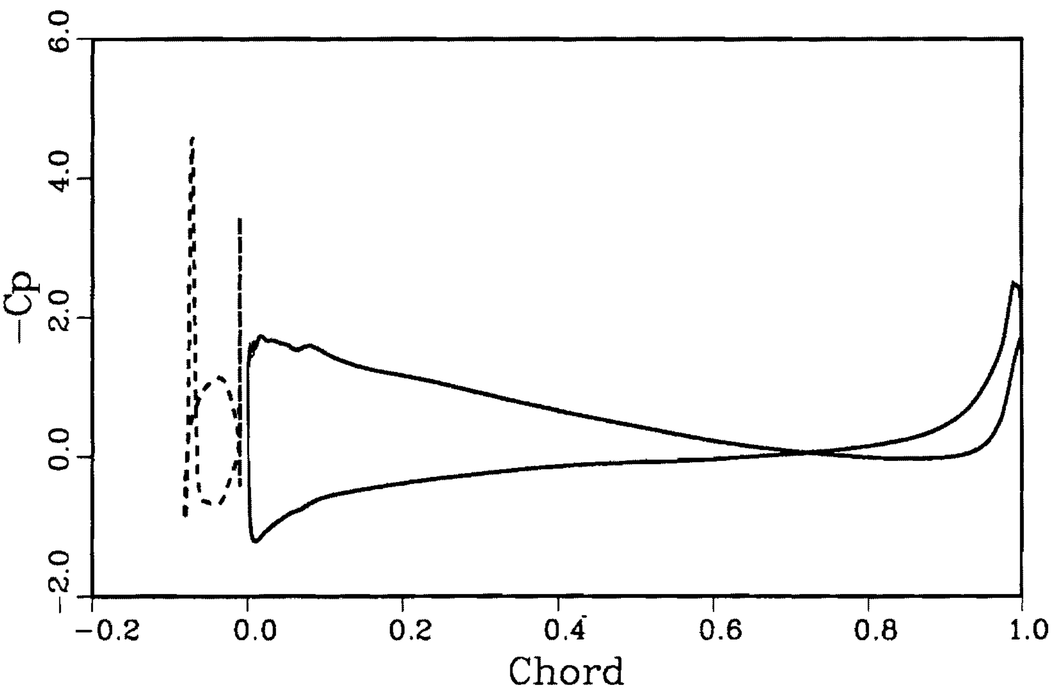


From Plot2





From plot3



B.7 Program Listing

```
      PROGRAM GEOM

C   **********************************************************************
C   *    This program performs conformal transformation of slat-airfoil  *
C   *     geometry in Z-plane to two-concentric circles in ZETA-plane    *
C   **********************************************************************

      PARAMETER (II=160,JJ=60,IIN=161,IC=120)
      PARAMETER (IIP1=161,IIP2=162)
      PARAMETER (KFC1=241,KFC2=242)
      COMPLEX Z,Z1,Z2,Z3,Z4,Z5,ZETA
      COMPLEX ZS,ZA,Z1S,Z1A,Z2S,Z2A,Z3S,Z3A
     %        ,Z4S,Z4A,Z5S,Z5A,Z6A,Z6S
      COMPLEX W1,W2,W3,W4,EPSS,EPSA,SS,SA,YC
      COMPLEX WW,WWT,Z1TS,Z1NS,Z1TA,Z1NA
      COMPLEX WROT
      COMPLEX RZ,ROZ,RZJ,ROZJ
      COMPLEX ZC,ZCBAR
      COMPLEX Z4INF,Z5INF,Z6INF
      COMPLEX DZDZE,DZDZ1,DZ1DZ2,DZ2DZ3,DZ3DZ4,DZ4DZ5,DZ5DZE
      COMMON/ZZ/Z3S(II),Z4S(II)
      COMMON/COEF/M3,A0,B0,A(IC),B(IC)
      COMMON/COEF1/M31,C(IC),D(IC)
      COMMON/RSS/R
      DIMENSION COEF(IIP1,2)
      DIMENSION UC(II,JJ),VC(II,JJ)
      DIMENSION ASS(KFC1),BSS(KFC1),CSS(KFC1),DSS(KFC1),
     %          ASA(KFC1),BSA(KFC1),CSA(KFC1),DSA(KFC1)
      DIMENSION UA(II,JJ),UB(II,JJ)
      DIMENSION H(II,JJ),R1(JJ),R2(JJ)
      DIMENSION Z(IIP1,JJ),Z1(IIP1,JJ),Z2(IIP1,JJ),Z3(IIP1,JJ),
     %          Z4(IIP1,JJ),Z5(IIP1,JJ),ZETA(IIP1,JJ)
      DIMENSION ZS(II),ZA(II),Z1S(II),Z1A(II),Z2S(II),Z2A(II)
      DIMENSION Z3A(II),Z4A(II),Z5S(II),Z5A(II),Z6S(II),Z6A(II)
      DIMENSION XX(IIP1,JJ),YY(IIP1,JJ),XW(IIP1,6),YW(IIP1,6)
      DIMENSION ACR(KFC1),BCR(KFC1),CCR(KFC1),DCR(KFC1)
      DIMENSION AC1(KFC1),BC1(KFC1),CC1(KFC1),DC1(KFC1)
      DIMENSION XA(IIP1),YA(IIP1),XS(IIP1),YS(IIP1)
      DIMENSION PHI(IIP1),RHO(IIP1),DTH(IIP1),DPH(IIP1)
      DIMENSION YC(IIP1),TH(IIP1),IO(6),JO(6)
      DIMENSION ALNRI(IIP2),THEIN(IIP2),THWORK(IIP2),ALWORK(IIP2)
      DIMENSION VRZE(II,JJ),VTHZE(II,JJ)
      DIMENSION VRW(II),VTHW(II)
      DIMENSION AM(II,II),BF(II),IPVT(II)
      DIMENSION CS(KFC1,IIP1),SN(KFC1,IIP1),RP(JJ,KFC2),RL(JJ)
      DIMENSION CSP(KFC1,IIP1),SNP(KFC1,IIP1)
      DIMENSION SGMA(KFC1)
      DIMENSION CGAR(KFC1),CGA1(KFC1)
      DIMENSION XXA(IIP1),YYA(IIP1)
      DIMENSION XT(II),YT(II),XB(II),YB(II)
      DIMENSION XXC(II,JJ),XXN(II,JJ)
      REAL KAPAA,KAPAS
      LOGICAL LB2
      DATA SCALE,XTIN,YTIN/1.,0.,0./
      DATA AL/3.6/
      DATA XROT/0.9/
C   -------------------------------------------------------------------------
C   |  M:     no. of grid points in PHI-direction                            |
C   |  N:     no. of grid points in RHO-direction, velocity grid             |
C   |  TAUS,TAUA:  trailing edge angle of slat and airfoil respectively      |
C   |  ERSR,EPSI:  complex number of coordinates mesuared from the slat      |
C   |              nose to the half of nose radius                           |
C   |  ERAR,EPAI:  complex number of coordinates mesuared from the airfoil   |
C   |              nose to the half of nose radius                           |
C   |  FACT:  factor value for grid plot                                     |
C   |  MSN:   input numbering of the slat nose coordinate                    |
```

```
C  |   AMP:   Amplification on PHI-direction                                  |
C  |   AMPA:  Amplification on RHO-direction from airfoil surface             |
C  |   AMPS:  Amplification on RHO-direction from slat surface                |
C  |   NA:  No. of grid spacing from airfoil before the singular point        |
C   ------------------------------------------------------------------------

      OPEN(UNIT=5,FILE='geom.in',STATUS='OLD',FORM='FORMATTED')
      OPEN(UNIT=7,FILE='dd',STATUS='OLD',FORM='FORMATTED')
      READ(5,*) M,N,M3,M31
      READ(5,*) TAUS,TAUA
      READ(5,*) EPSR,EPSI,EPAR,EPAI
      READ(5,*) MSN,MAN
      READ(5,*) AMP,AMPA,AMPS
      READ(5,*) NA,UR,KFC,LB2

      MM1=M-1

      EPSS=CMPLX(EPSR,EPSI)
      EPSA=CMPLX(EPAR,EPAI)
C   -------------------------------------------------------
C  | XS,YS: x- and y-coordinates of input slat geometry     |
C  | XA,YA: x- and y-coordinates of input airfoil geometry  |
C   -------------------------------------------------------
      READ(7,*) MS,MA
      DO 151 I=1,MS
      READ(7,*) XXA(I),YYA(I)
  151 CONTINUE

C.... Add rotation of the front part here !!!!!!!!

      DO 153 I=1,MS
      XS(I)=XXA(MS+1-I)
      YS(I)=YYA(MS+1-I)
  153 CONTINUE

      DO 152 I=1,MA
      READ(7,*) XXA(I),YYA(I)
  152 CONTINUE
      do 159 i=1,ma
      xa(i)=xxa(ma+1-i)
      ya(i)=yya(ma+1-i)
  159 continue
      DO 155 I=1,MS
      XS(I)=((XS(I))*SCALE+XTIN)*AL
      YS(I)=(YS(I)*SCALE+YTIN)*AL
  155 CONTINUE
      DO 156 I=1,MA
      XA(I)=XA(I)*AL
      YA(I)=YA(I)*AL
  156 CONTINUE

      DO 110 I=1,MS
      ZS(I)=CMPLX(XS(I),YS(I))
  110 CONTINUE
      DO 111 I=1,MA
      ZA(I)=CMPLX(XA(I),YA(I))
  111 CONTINUE

      PI=3.1415926535898

      IF(LB2) THEN

C ---  expand slat

      ZITS=ZS(1)
      ZINS=ZS(MSN)-EPSS
```

```
      KAPAS=2.-TAUS/PI
      SS=(Z1TS-Z1NS)/(2.*KAPAS)
      RKAPAS=1./KAPAS

C     PRINT*,' Z1TS Z1NS KAPAS SS = ',Z1TS,Z1NS,KAPAS,SS

      DO 10 I=2,MS-1
      W1=ZS(I)-Z1TS
      W2=ZS(I)-Z1NS
      W3=(W1/W2)
      WW1=REAL(W3)
      WW2=AIMAG(W3)
      RHO(I)=SQRT(WW1*WW1+WW2*WW2)
      PHI(I)=ATAN2(WW2,WW1)
   10 CONTINUE
      I1=MS
      DO 121 I=2,MS-1
      DPHI=ABS(PHI(I+1)-PHI(I))
      IF (DPHI.GE.PI) THEN
      I1=I+1
      GO TO 122
      ENDIF
  121 CONTINUE
  122 CONTINUE
      DO 123 I=I1,MS
      PHI(I)=PHI(I)-2.*PI
  123 CONTINUE
      I1=MS
      DO 131 I=2,MS-1
      DPHI=ABS(PHI(I+1)-PHI(I))
      IF (DPHI.GE.PI) THEN
      I1=I+1
      GO TO 132
      ENDIF
  131 CONTINUE
  132 CONTINUE
      DO 138 I=I1,MS
      PHI(I)=PHI(I)+2.*PI
  138 CONTINUE
      Z1S(1)=SS
      Z1S(MS)=SS
      DO 11 I=2,MS
      RHO2=(RHO(I))**RKAPAS
      THE2=RKAPAS*PHI(I)
      AX=RHO2*COS(THE2)
      AY=RHO2*SIN(THE2)
      W3=CMPLX(AX,AY)
      Z1S(I)=SS*(1.+W3)/(1.-W3)
   11 CONTINUE
      DO 13 I=1,MA
      W1=ZA(I)-Z1TS
      W2=ZA(I)-Z1NS
      W3=(W1/W2)**RKAPAS
      Z1A(I)=SS*(1.+W3)/(1.-W3)
   13 CONTINUE
      ELSE
       DO 133 I=1,MA
        Z1A(I)=ZA(I)
  133 CONTINUE
      ENDIF

      if(.not.lb2) then
      do 135 i=1,ms
      z1s(i)=0.
  135 continue
      endif
```

```
c -- expand airfoil

      ZITA=Z1A(1)
      IF(LB2) THEN
      Z1NA=Z1A(MAN)-EPSA
      ELSE
      z1na=z1a(man)-epsa
      ENDIF
      KAPAA=2.-TAUA/PI
      SA=(Z1TA-Z1NA)/(2.*KAPAA)
      RKAPAA=1./KAPAA
C
      DO 20 I=2,MA-1
      W1=Z1A(I)-Z1TA
      W2=Z1A(I)-Z1NA
      W3=(W1/W2)
      WW1=REAL(W3)
      WW2=AIMAG(W3)
      RHO(I)=SQRT(WW1*WW1+WW2*WW2)
      PHI(I)=ATAN2(WW2,WW1)
   20 CONTINUE
      I1=MA
      DO 221 I=2,MA-1
      DPHI=ABS(PHI(I+1)-PHI(I))
      IF (DPHI.GE.PI) THEN
      I1=I+1
      GO TO 222
      ENDIF
  221 CONTINUE
  222 CONTINUE
      DO 223 I=I1,MA
      PHI(I)=PHI(I)+2.*PI
  223 CONTINUE
      I1=MA
      DO 321 I=2,MA-1
      DPHI=ABS(PHI(I+1)-PHI(I))
      IF (DPHI.GE.PI) THEN
      I1=I+1
      GO TO 322
      ENDIF
  321 CONTINUE
  322 CONTINUE
      DO 323 I=I1,MA
      PHI(I)=PHI(I)-2.*PI
  323 CONTINUE

      Z2A(1)=SA
      Z2A(MA)=SA
      DO 23 I=2,MA
      RHO2=(RHO(I))**RKAPAA
      THE2=RKAPAA*PHI(I)
      AX=RHO2*COS(THE2)
      AY=RHO2*SIN(THE2)
      W3=CMPLX(AX,AY)
      Z2A(I)=SA*(1.+W3)/(1.-W3)
   23 CONTINUE

      IF(LB2) THEN
      DO 21 I=1,MS
      W1=Z1S(I)-Z1TA
      W2=Z1S(I)-Z1NA
      W3=(W1/W2)**RKAPAA
      Z2S(I)=SA*(1.+W3)/(1.-W3)
   21 CONTINUE
      ENDIF
```

```
C  -- Prepare input coordinates of the airfoil for Theodosen Transformation

      DO 231 I=1,MA-1
      WR=REAL(Z2A(I))
      WI=AIMAG(Z2A(I))
      WA=SQRT(WR*WR+WI*WI)
      ALNRI(I)=ALOG(WA)
      THEIN(I)=ATAN2(WI,WR)
  231 CONTINUE
      THET=THEIN(1)
      DO 241 I=2,MA-1
      DPHI=ABS(THEIN(I+1)-THEIN(I))
      IF (DPHI.GE.PI) THEN
      I1=I+1
      GO TO 242
      ENDIF
  241 CONTINUE
  242 CONTINUE
      DO 243 I=I1,MA
      THEIN(I)=THEIN(I)+2.*PI
  243 CONTINUE
      ALNRI(MA)=ALNRI(1)
      THEIN(MA)=THEIN(1)+2.*PI
      DO 291 I=1,MA
      IF(THEIN(I).LT.0.) THEIN(I)=THEIN(I)+2.*PI
      IF(THEIN(I).GT.2.*PI) THEIN(I)=THEIN(I)-2.*PI
  291 CONTINUE
      DO 2911 I=1,MA
      THWORK(I)=THEIN(I)
      ALWORK(I)=ALNRI(I)
 2911 CONTINUE
      I1=1
      DO 292 I=1,MA-1
      DPHI=ABS(THEIN(I+1)-THEIN(I))
      IF(DPHI.GE.PI) THEN
      I1=I+1
      GO TO 293
      ENDIF
  292 CONTINUE
  293 CONTINUE
      IF(I1.EQ.MA) I1=1
c       PRINT*,' I1 ',I1
      DO 294 I=I1,MA-1
      I2=I-I1+2
      THEIN(I2)=THWORK(I)
      ALNRI(I2)=ALWORK(I)
  294 CONTINUE
      DO 295 I=1,I1-1
      I3=I2+I
      THEIN(I3)=THWORK(I)
      ALNRI(I3)=ALWORK(I)
  295 CONTINUE
      THEIN(MA+1)=THEIN(2)+2.*PI
      ALNRI(MA+1)=ALNRI(2)
      THEIN(1)=THEIN(MA)-2.*PI
      ALNRI(1)=ALNRI(MA)

C --- Theodorsen Tansformation

      CALL THDSN(ALNRI,THEIN,MA,THET)

C  -- only airfoil trailing edge information are needed to be
C      carried on from this point on

c      Print*,' z1a z2a z3a ',z1a(1),z2a(1),z3a(1)
```

```
c      print*,' at Z2A(1) ',z2a(1)

      CALL NEWTON(Z2A(1),Z3A(1))
       w3=cmplx(a0,b0)
       do 4044 jc=1,m3
      w1=cmplx(a(jc),b(jc))
       w3=w3+w1/z3a(1)**(jc)
 4044 continue
        w4=exp(w3)
       z2a(1)=z3a(1)*w4

c        print*,' z3a(1) z2a(1) = ',z3a(1),z2a(1)

      IF(LB2) THEN
      DO 252 I=1,MS-1
      CALL NEWTON(Z2S(I),Z3S(I))
  252 CONTINUE
      Z3S(MS)=Z3S(1)

C  --- Bilinear transformation

      CALL FIND(ZC,MS,UR)

      WR=REAL(ZC)
      WI=AIMAG(ZC)
      ZCBAR=CMPLX(WR,-WI)
      ZCABS=CABS(ZC)

C  --- for slat

      DO 501 I=1,MS-1
      W1=(Z3S(I)*ZCBAR-ZCABS*ZCABS)/(Z3S(I)*ZCBAR-1.)
      Z4S(I)=W1/ZCABS
  501 CONTINUE
      Z4S(MS)=Z4S(1)

C  -- for the airfoil trailing edge

       I=1
      W1=(Z3A(I)*ZCBAR-ZCABS*ZCABS)/(Z3A(I)*ZCBAR-1.)
      Z4A(I)=W1/ZCABS

      DO 631 I=1,MS-1
      WR=REAL(Z4S(I))
      WI=AIMAG(Z4S(I))
      WA=SQRT(WR*WR+WI*WI)
      ALNRI(I)=ALOG(WA)
      THEIN(I)=ATAN2(WI,WR)
  631 CONTINUE
      DO 641 I=2,MS-1
      DPHI=ABS(THEIN(I+1)-THEIN(I))
      IF (DPHI.GE.PI) THEN
      I1=I+1
      GO TO 642
      ENDIF
  641 CONTINUE
  642 CONTINUE
      DO 643 I=I1,MS
      THEIN(I)=THEIN(I)+2.*PI
  643 CONTINUE
      ALNRI(MS)=ALNRI(1)
      THEIN(MS)=THEIN(1)+2.*PI

      DO 691 I=1,MS
      IF(THEIN(I).LT.0.) THEIN(I)=THEIN(I)+2.*PI
      IF(THEIN(I).GT.2.*PI) THEIN(I)=THEIN(I)-2.*PI
```

```
  691 CONTINUE
      DO 6911 I=1,MS
      THWORK(I)=THEIN(I)
      ALWORK(I)=ALNRI(I)
 6911 CONTINUE
      I1=1
      DO 692 I=1,MS-1
      DPHI=ABS(THEIN(I+1)-THEIN(I))
      IF(DPHI.GE.PI) THEN
      I1=I+1
      GO TO 693
      ENDIF
  692 CONTINUE
  693 CONTINUE
      IF(I1.EQ.MS) I1=1
c      PRINT*,' I1 = ', I1
      DO 694 I=I1,MS-1
      I2=I-I1+2
      THEIN(I2)=THWORK(I)
      ALNRI(I2)=ALWORK(I)
  694 CONTINUE
      DO 695 I=1,I1-1
      I3=I2+I
      THEIN(I3)=THWORK(I)
      ALNRI(I3)=ALWORK(I)
  695 CONTINUE
      THEIN(MS+1)=THEIN(2)+2.*PI
      ALNRI(MS+1)=ALNRI(2)
      THEIN(1)=THEIN(MS)-2.*PI
      ALNRI(1)=ALNRI(MS)
      CALL GARCK(ALNRI,THEIN,MS)

C  ---  only the slat trailing coordinate information are needed
C        from point on

      CALL NEWTON1(Z4S(1),Z5S(1))

c      PRINT*,' Z4S = ',Z4S(1)
c      PRINT*,' Z5S = ',Z5S(1)

      CALL NEWTON1(Z4A(1),Z5A(1))

c      PRINT*,' Z4A Z5A(1) = ',Z4A(1),Z5A(1)

C  --- point of infinity is to be transformed to a point on positive
C        real axis

      Z4INF=CMPLX(1./ZCABS,0.)
      CALL NEWTON1(Z4INF,Z5INF)

C  --- to find rotate angle

      WR=REAL(Z5INF)
      WI=AIMAG(Z5INF)
      THINF=ATAN2(WI,WR)
      IF(THINF.LT.0.) THINF=THINF+2.*PI
      THROT=2.*PI-THINF
      ELSE
       Z4A(1)=-1./Z3A(1)
       Z5A(1)=Z4A(1)
       Z4INF=CMPLX(0.,0.)
       Z5INF=Z4INF
       THROT=0.
      ENDIF
      W1=CMPLX(0.,THROT)
      WROT=EXP(W1)
```

```
      Z6INF=Z5INF*WROT
      Z6A(1)=Z5A(1)*WROT
      Z6S(1)=Z5S(1)*WROT

c      PRINT*,' THINF THROT ',THINF,THROT
c      PRINT*,' *** Z6INF  = ',Z6INF
c      PRINT*,' *** Z5INF  = ',Z5INF
c      PRINT*,' *** Z4INF  = ',Z4INF
c      PRINT*,' *** Z6A    = ',Z6A(1)
c      PRINT*,' *** Z6S    = ',Z6S(1)

      WR=REAL(Z6A(1))
      WI=AIMAG(Z6A(1))
      TH6A=ATAN2(WI,WR)
      IF(TH6A.LT.O.) TH6A=TH6A+2.*PI
      WROT=EXP(-WI)
      PRINT*,'  ** AIRFORIL TRAILING EDGE ANGLE = ',TH6A
      if(LB2) then
      th6s=atan2(aimag(z6s(1)),real(z6s(1)))
      if(th6s.lt.0.) th6s=th6s+2.*pi
      print*,'  ** SLAT TRAILING EDGE ANGLE = ',th6s
      endif

C  -- Start to tranform back from ZETA

C ****  ZETA(I,1) denotes slat, ZETA(I,N) denotes airfoil

c      PRINT*,' AMP = ',AMP

      MC=(M-1)/2

C  -- Construct a variable PHI grid

      IF(AMP.EQ.1.) THEN
       WA=FLOAT(MC)
        ELSE
        WA=(AMP**MC-1.)/(AMP-1.)
       ENDIF
      DTH1=PI/WA
      DTH(1)=DTH1
      DO 77 I=2,MC
      DTH(I)=DTH(I-1)*AMP
   77 CONTINUE
      PHI(1)=0.
      DO 78 I=2,MC+1
      PHI(I)=PHI(I-1)+DTH(I-1)
   78 CONTINUE
      DO 79 I=MC+2,M
      PHI(I)=2.*PI-PHI(M+1-I)
   79 CONTINUE

      do 80 i=1,m
      phi(i)=phi(i)+th6a
      if(Phi(i).gt.2.*pi) phi(i)=phi(i)-2.*pi
   80 continue

      if(LB2) then
c      print*,' th6a  th6s ** ',th6a,th6s
      do 71 i=2,m-1
      if(th6s.le.phi(i).and.th6s.gt.phi(i-1)) go to 72
   71 continue
   72 i1=i

       dph1=th6s-phi(i-1)
       dph2=phi(i)-th6s
       if(dph2.ge.dph1) then
```

```
         phi(i-1)=th6s
         phi(i-2)=.5*(phi(i-3)+phi(i-1))
         phi(i)=.5*(phi(i+1)+phi(i-1))
        else
        phi(i1)=th6s
        phi(i1-1)=.5*(phi(i1)+phi(i1-2))
        phi(i1+1)=.5*(phi(i1)+phi(i1+2))
        endif
c        print*,' ** i1 phi(i1-2) phi(i1+2) ',i1,(phi(i),i=i1-2,i1+2)
        endif

C  -- Construct a uniform PHI-grid

        DPHI=2.*PI/FLOAT(M-1)
        TH(1)=0.
        DO 1001 I=2,M
        TH(I)=TH(I-1)+DPHI
 1001   CONTINUE

C  -- Construct grid on RHO-direction

        RAP=REAL(Z6INF)
        IF(.NOT.LB2) THEN
         R=0.1
         RAP=R
        ENDIF
        WA=(.5*(AMPA**NA)*(1.+AMPA)-1.)/(AMPA-1.)
        DA1=(1.-RAP)/WA
        DA=DA1
        R1(1)=R
        R1(N)=1.
        DO 1007 J=N-1,N-NA,-1
        R1(J)=R1(J+1)-DA
        DA=DA*AMPA
 1007   CONTINUE
        NS=N-NA-2
        WA=(.5*(AMPS**NS)*(1.+AMPS)-1.)/(AMPS-1.)
        DA1=(RAP-R)/WA
        DA=DA1
        DO 1008 J=2,NS+1
        R1(J)=R1(J-1)+DA
        DA=DA*AMPS
 1008   CONTINUE
        DO 1006 J=1,N-1
        R2(J)=.5*(R1(J)+R1(J+1))
 1006   CONTINUE

        DO 1003 J=1,N-1
        DO 1002 I=1,M
        ZETA(I,J)=CMPLX(R2(J)*COS(PHI(I)),R2(J)*SIN(PHI(I)))
 1002   CONTINUE
 1003   CONTINUE

C--- Rotate

        DO 1010 J=1,N-1
        DO 1010 I=1,M
        Z5(I,J)=ZETA(I,J)*WROT
 1010   CONTINUE

C  -- Inverse Garrick Transformation

        IF(LB2) THEN
        DO 1021 J=1,N-1
        DO 1021 I=1,M
        W3=0.
```

```
      DO 1020 JC=1,M31
      W1=CMPLX(-C(JC),D(JC))
      W2=CMPLX(C(JC),D(JC))
      RZ=R*Z5(I,J)
      ROZ=R/Z5(I,J)
      RZJ=RZ**(JC)
      ROZJ=ROZ**(JC)
      W3=W3+W1*RZJ+W2*ROZJ
 1020 CONTINUE
      W4=EXP(W3)
      Z4(I,J)=Z5(I,J)*W4
 1021 CONTINUE

C-- Inverse Bilinear Transformation

      DO 1031 J=1,N-1
      DO 1031 I=1,M
      W1=ZCABS-Z4(I,J)
      W2=-Z4(I,J)*ZCABS+1.
      Z3(I,J)=(ZCABS*W1)/(ZCBAR*W2)
 1031 CONTINUE
      ELSE
        DO 1022 J=1,N-1
        DO 1022 I=1,M
        Z4(I,J)=Z5(I,J)
 1022   CONTINUE
        DO 1032 J=1,N-1
        DO 1032 I=1,M
      Z3(I,J)=-1./Z4(I,J)
 1032   CONTINUE
      ENDIF

C  -- Inverse Theodosen Transformation

      DO 1040 J=1,N-1
      DO 1040 I=1,M
      W3=CMPLX(A0,B0)
      DO 1041 JC=1,M3
      W1=CMPLX(A(JC),B(JC))
      W3=W3+W1/Z3(I,J)**(JC)
 1041 CONTINUE
      W4=EXP(W3)
      Z2(I,J)=Z3(I,J)*W4
 1040 CONTINUE

C --- Inverse Karman-Trefftz Airfoil

      DO 1051 J=1,N-1
      DO 1051 I=1,M
      W1=Z2(I,J)-SA
      W2=Z2(I,J)+SA
      W3=(W1/W2)**KAPAA
      Z1(I,J)=(Z1TA-W3*Z1NA)/(1.-W3)
 1051 CONTINUE

C  -Inverse Karman-Trefftz for slat

       IF(LB2) THEN
      DO 1061 J=1,N-1
      DO 1061 I=1,M
      W1=Z1(I,J)-SS
      W2=Z1(I,J)+SS
      W3=(W1/W2)**KAPAS
      Z(I,J)=(Z1TS-W3*Z1NS)/(1.-W3)
 1061 CONTINUE
      ELSE
```

```
       DO 1062 J=1,N-1
       DO 1062 I=1,M
       Z(I,J)=Z1(I,J)
 1062 CONTINUE
      ENDIF

C  -- Compute the scale factor

      DO 1501 J=1,N-1
      DO 1501 I=1,M-1
      DZ5DZE=WROT
      dz4dz5=1.
      dz3dz4=1./(z4(i,j)**2)
      IF(LB2) THEN
      W3=0.
      DO 1601 JC=1,M31
      W1=CMPLX(-C(JC),D(JC))
      W2=CMPLX(C(JC),D(JC))
      RZ=R*Z5(I,J)
      ROZ=R/Z5(I,J)
      RZJ=RZ**(JC)
      ROZJ=ROZ**(JC)
      W3=W3+W1*FLOAT(JC)*RZJ/Z5(I,J)
     %     -W2*FLOAT(JC)*ROZJ/Z5(I,J)
 1601 CONTINUE
      DZ4DZ5=Z4(I,J)/Z5(I,J)+Z4(I,J)*W3

      W1=(Z4(I,J)*ZCABS-1.)**2
      DZ3DZ4=ZCABS*(ZCABS**2-1.)/(ZCBAR*W1)

      ENDIF
      W3=0.
      DO 1701 JC=1,M3
      W1=CMPLX(A(JC),B(JC))
      W3=W3-W1*FLOAT(JC)/Z3(I,J)**(JC+1)
 1701 CONTINUE
      DZ2DZ3=Z2(I,J)/Z3(I,J)+Z2(I,J)*W3

      W1=((Z2(I,J)-SA)/(Z2(I,J)+SA))**(KAPAA-1.)
      W2=(Z1(I,J)-Z1NA)**2/(Z1TA-Z1NA)
      DZ1DZ2=KAPAA*W2*W1*2.*SA/(Z2(I,J)+SA)**2

      dzdz1=1.
      IF(LB2) THEN
      W1=((Z1(I,J)-SS)/(Z1(I,J)+SS))**(KAPAS-1.)
      W2=(Z(I,J)-Z1NS)**2/(Z1TS-Z1NS)
      DZDZ1=KAPAS*W2*W1*2.*SS/(Z1(I,J)+SS)**2
      ENDIF
      DZDZE=DZDZ1*DZ1DZ2*DZ2DZ3*DZ3DZ4*DZ4DZ5*DZ5DZE
      H(I,J)=CABS(DZDZE)**2
 1501 CONTINUE

c     PRINT*,' ***** THE SCALE FACTOR **** '
c     DO 1599 J=1,N-1
c     PRINT*,'  J =  ',J
c     WRITE(6,100)  (H(I,J),I=1,M-1)
c1599 CONTINUE
c  100 FORMAT(1X,10F8.3)

C -- Compute smooth function

      DDTH=PI/FLOAT(KFC-1)
      DO 3211 K=2,KFC
      THE=FLOAT(K-1)*DDTH
      SGMA(K)=SIN(THE)/THE
 3211 CONTINUE
```

```
C  -- Contruct COS and SIN matrix

      DO 3202 K=1,KFC
      DO 3202 I=1,M-1
      THE=FLOAT(K-1)*TH(I)
      CS(K,I)=COS(THE)
      SN(K,I)=SIN(THE)
 3202 CONTINUE
      DO 3203 K=1,KFC
      CS(K,M)=CS(K,1)
      SN(K,M)=SN(K,1)
      CSP(K,M)=CSP(K,1)
      SNP(K,M)=SNP(K,1)
 3203 CONTINUE
      DO 3208 K=1,KFC
      DO 3208 I=1,M-1
      THE=FLOAT(K-1)*PHI(I)
      CSP(K,I)=COS(THE)
      SNP(K,I)=SIN(THE)
 3208 CONTINUE
      DO 3209 K=1,KFC
      CSP(K,M)=CSP(K,1)
      SNP(K,M)=SNP(K,1)
 3209 CONTINUE

C  -- Construct array RP and RL

      DO 3221 J=1,N
      RP(J,1)=R1(J)
      RL(J)=ALOG(R1(J))
      DO 3221 K=2,KFC+1
      RP(J,K)=RP(J,K-1)*RP(J,1)
 3221 CONTINUE

C --  Write out

      WRITE(2) AL,th6s,na,ns
      WRITE(2) H
      WRITE(2) CSP,SNP
      WRITE(2) RP,RL,SGMA
      WRITE(2) R1,R2
      WRITE(2) TH,PHI

C -- CONSTRUCT PHI AND DPH ARRAY

      PHIMIN=PHI(1)
      II2=1
      DO 99 I=2,M-1
      IF(PHI(I).LT.PHIMIN) THEN
      II2=I
      PHIMIN=PHI(I)
      ENDIF
   99 CONTINUE
      ISS=II2-2
      IF(ISS.LT.0) ISS=ISS+M-1
      DO 995 I=II2,M-1
      PHI(I)=PHI(I)+2.*PI
  995 CONTINUE
      PHI(M)=PHI(1)+2.*PI

C   Construct DPH array

      DO 999 I=2,M-1
      DPH(I)=.5*(PHI(I+1)-PHI(I-1))
  999 CONTINUE
```

```
      DPH(1)=.5*(PHI(2)+2.*PI-PHI(M-1))

      DO 2003 J=1,N
      DO 2002 I=1,M
      ZETA(I,J)=CMPLX(R1(J)*COS(PHI(I)),R1(J)*SIN(PHI(I)))
 2002 CONTINUE
 2003 CONTINUE

C  -- Determine velocity contributions from the singular point-------
C     This is to get Fourier Coeff. of the singular contribution from a uniform
C     grid on PHI-direction

      RAP=REAL(Z6INF)
      RA=RAP
      IF(LB2) RA=1./ZCABS
      PRINT*,' RAP RA ',RAP,RA
      RA2=RA*RA
      RAP2=RAP*RAP
      ww=-exp(cmplx(a0,b0))
      IF(LB2) THEN
      W3=0.
      DO 2201 JC=1,M31
      W1=CMPLX(-C(JC),D(JC))
      W2=CMPLX(C(JC),D(JC))
      RZ=R*Z5INF
      ROZ=R/Z5INF
      RZJ=RZ**(JC)
      ROZJ=ROZ**(JC)
      W3=W3+W1*FLOAT(JC)*RZJ/Z5INF
     %      -W2*FLOAT(JC)*ROZJ/Z5INF
 2201 CONTINUE
      W1=EXP(CMPLX(AO,BO)) * RAP
      W4=(W3*Z5INF+1.)
      WW=(1.-ZCABS*ZCABS)*W1/(ZCBAR*W4)
      ENDIF
      WR=REAL(-WW)
      WI=AIMAG(-WW)
      H45=SQRT(WR*WR+WI*WI)
      THE45=ATAN2(WI,WR)
      COSTAU=COS(-THE45)
      SINTAU=SIN(-THE45)

      DO 2231 J=1,N,N-1
      DO 2231 I=1,M-1
      WR=REAL(ZETA(I,J))
      WI=AIMAG(ZETA(I,J))
      RHO1=SQRT(WR**2+WI**2)
      RHO12=RHO1*RHO1
      PHI1=ATAN2(WI,WR)
      CSPHI=COS(PHI1)
      SNPHI=SIN(PHI1)
      WA=RHO12+RAP2-2.*RAP*RHO1*CSPHI
      WA2=WA*WA
      WB=H45
      WC=(RAP2-RHO12)*SNPHI/WA2
      WD=((RAP2+RHO12)*CSPHI-2.*RAP*RHO1)/WA2
      UA(I,J)=WB*(COSTAU*WD+SINTAU*WC)
      UB(I,J)=WB*(COSTAU*WC-SINTAU*WD)
 2231 CONTINUE

C  -- Determine Fourier Coeff. of singular velocity contributions

      IF(LB2) THEN
      J=1
      DO 2399 I=1,M-1
      THWORK(I)=UA(I,J)
```

```
      ALWORK(I)=UB(I,J)
 2399 CONTINUE
      THWORK(M)=THWORK(1)
      ALWORK(M)=ALWORK(1)
      DO 2401 K=1,KFC
      ACR(K)=0.
      BCR(K)=0.
      CCR(K)=0.
      DCR(K)=0.
 2401 CONTINUE
      DO 2402 I=1,M-1
      IP1=I+1
      DDPH=DPH(I)
      FA=.5*(THWORK(I)+THWORK(IP1))
      WAA=FA*DDPH
      ACR(1)=ACR(1)+WAA/PI
      DO 2403 K=2,KFC
      FAC1=1./FLOAT(K-1)
      WAA= FA*FAC1*(SNP(K,IP1)-SNP(K,I))
      WBB=-FA*FAC1*(CSP(K,IP1)-CSP(K,I))
      ACR(K)=ACR(K)+WAA/PI
      BCR(K)=BCR(K)+WBB/PI
 2403 CONTINUE
      FA=.5*(ALWORK(I)+ALWORK(IP1))
      WAA=FA*DDPH
      CCR(1)=CCR(1)+WAA/PI
      DO 2404 K=2,KFC
      FAC1=1./FLOAT(K-1)
      WAA= FA*FAC1*(SNP(K,IP1)-SNP(K,I))
      WBB=-FA*FAC1*(CSP(K,IP1)-CSP(K,I))
      CCR(K)=CCR(K)+WAA/PI
      DCR(K)=DCR(K)+WBB/PI
 2404 CONTINUE
 2402 CONTINUE

C     PRINT*,' *** J = ',J
C     DO 2404 K=1,MC+1
C     WRITE(6,101)  K,TCR(K),SCR(K)
C2404 CONTINUE

      ENDIF

      J=N
      DO 2398 I=1,M-1
      THWORK(I)=UA(I,J)
      ALWORK(I)=UB(I,J)
 2398 CONTINUE
      THWORK(M)=THWORK(1)
      ALWORK(M)=ALWORK(1)
      DO 2411 K=1,KFC
      AC1(K)=0.
      BC1(K)=0.
      CC1(K)=0.
      DC1(K)=0.
 2411 CONTINUE
      DO 2412 I=1,M-1
      IP1=I+1
      DDPH=DPH(I)
      FA=.5*(THWORK(I)+THWORK(IP1))
      WAA=FA*DDPH
      AC1(1)=AC1(1)+WAA/PI
      DO 2413 K=2,KFC
      FAC1=1./FLOAT(K-1)
      WAA= FA*FAC1*(SNP(K,IP1)-SNP(K,I))
      WBB=-FA*FAC1*(CSP(K,IP1)-CSP(K,I))
      AC1(K)=AC1(K)+WAA/PI
```

```
      BC1(K)=BC1(K)+WBB/PI
 2413 CONTINUE
      FA=.5*(ALWORK(I)+ALWORK(IP1))
      WAA=FA*DDPH
      CC1(1)=CC1(1)+WAA/PI
      DO 2414 K=2,KFC
      FAC1=1./FLOAT(K-1)
      WAA= FA*FAC1*(SNP(K,IP1)-SNP(K,I))
      WBB=-FA*FAC1*(CSP(K,IP1)-CSP(K,I))
      CC1(K)=CC1(K)+WAA/PI
      DC1(K)=DC1(K)+WBB/PI
 2414 CONTINUE
 2412 CONTINUE

C  -- Determine velocity contribution from GAMAS

      IF(LB2) THEN
      J=1
      DO 2573 I=1,M-1
      CS1=COS(PHI(I))
      SN1=SIN(PHI(I))
      WA=R1(J)**2+RAP2 -2.*RAP*R1(J)*CS1
      THWORK(I)=(R1(J)-RAP*CS1)/WA
 2573 CONTINUE
      THWORK(M)=THWORK(1)
      DO 2501 K=1,KFC
      CGAR(K)=0.
 2501 CONTINUE
      DO 2502 I=1,M-1
      IP1=I+1
      DDPH=DPH(I)
      FA=.5*(THWORK(I)+THWORK(IP1))
      WAA=FA*DDPH
c     WAA=FA*DDPH+.5*FB*DDPH*(PHI(IP1)+PHI(I))
      CGAR(1)=CGAR(1)+WAA/PI
      DO 2503 K=2,KFC
      FAC1=1./FLOAT(K-1)
      WAA= FA*FAC1*(SNP(K,IP1)-SNP(K,I))
      CGAR(K)=CGAR(K)+WAA/PI
 2503 CONTINUE
 2502 CONTINUE
      ENDIF
      J=N
      DO 2673 I=1,M-1
      CS1=COS(PHI(I))
      SN1=SIN(PHI(I))
      WA=R1(J)**2+RAP2 -2.*RAP*R1(J)*CS1
      THWORK(I)=(R1(J)-RAP*CS1)/WA
 2673 CONTINUE
      THWORK(M)=THWORK(1)
      DO 2601 K=1,KFC
      CGA1(K)=0.
 2601 CONTINUE
      DO 2602 I=1,M-1
      IP1=I+1
      DDPH=DPH(I)
      FA=.5*(THWORK(I)+THWORK(IP1))
      WAA=FA*DDPH
      CGA1(1)=CGA1(1)+WAA/PI
      DO 2603 K=2,KFC
      FAC1=1./FLOAT(K-1)
      WAA= FA*FAC1*(SNP(K,IP1)-SNP(K,I))
      CGA1(K)=CGA1(K)+WAA/PI
 2603 CONTINUE
 2602 CONTINUE
  919 FORMAT(8F10.3)
```

```
      DO 2671 I=1,M
      THWORK(I)=.5*CGA1(1)
      DO 2672 K=2,KFC
      THWORK(I)=THWORK(I)+CGA1(K)*CSP(K,I)
 2672 CONTINUE
 2671 CONTINUE

C-- Now to construct the stretch-grid on PHI-direction

      DO 2083 J=1,N
      DO 2082 I=1,M
      ZETA(I,J)=CMPLX(R1(J)*COS(PHI(I)),R1(J)*SIN(PHI(I)))
 2082 CONTINUE
 2083 CONTINUE

C -- The following is to check the singular velocity at each grid point from
C      the Fourier Coeff. compared with values that are direct evaluated

      DO 2281 J=1,N
      DO 2281 I=1,M-1
      WR=REAL(ZETA(I,J))
      WI=AIMAG(ZETA(I,J))
      RHO1=SQRT(WR**2+WI**2)
      RHO12=RHO1*RHO1
      PHI1=ATAN2(WI,WR)
      CSPHI=COS(PHI1)
      SNPHI=SIN(PHI1)
      WA=RHO12+RAP2-2.*RAP*RHO1*CSPHI
      WA2=WA*WA
      WB=H45
      WC=(RAP2-RHO12)*SNPHI/WA2
      WD=((RAP2+RHO12)*CSPHI-2.*RAP*RHO1)/WA2
      UA(I,J)=WB*(costau*WD+sintau*WC)
      UB(I,J)=WB*(costau*WC-sintau*WD)
 2281 CONTINUE

  101 FORMAT(I4,4F10.6)

      NSING=N-NA-1

      WRITE(2)  ACR,BCR,CCR,DCR
      WRITE(2)  AC1,BC1,CC1,DC1
      WRITE(2)  RAP,CGAR,CGA1
      WRITE(2)  UA,UB

C  WRITE DATA FOR PLOT STREAMLINES AND VORTICITY CONTOURS

      DO 4003 J=1,N
      DO 4002 I=1,M
      ZETA(I,J)=CMPLX(R1(J)*COS(PHI(I)),R1(J)*SIN(PHI(I)))
 4002 CONTINUE
 4003 CONTINUE

C--- Rotate

      DO 4010 J=1,N
      DO 4010 I=1,M
      Z5(I,J)=ZETA(I,J)*WROT
 4010 CONTINUE

C  -- Inverse Garrick Transformation

      IF(LB2) THEN
      DO 4021 J=1,N
      DO 4021 I=1,M
      W3=0.
```

```
      DO 4020 JC=1,M31
      W1=CMPLX(-C(JC),D(JC))
      W2=CMPLX(C(JC),D(JC))
      RZ=R*Z5(I,J)
      ROZ=R/Z5(I,J)
      RZJ=RZ**(JC)
      ROZJ=ROZ**(JC)
      W3=W3+W1*RZJ+W2*ROZJ
 4020 CONTINUE
      W4=EXP(W3)
      Z4(I,J)=Z5(I,J)*W4
 4021 CONTINUE

C-- Inverse Bilinear Transformation

      DO 4031 J=1,N
      DO 4031 I=1,M
      W1=ZCABS-Z4(I,J)
      W2=-Z4(I,J)*ZCABS+1.
      Z3(I,J)=(ZCABS*W1)/(ZCBAR*W2)
 4031 CONTINUE
      ELSE
       DO 4022 J=1,N
       DO 4022 I=1,M
       Z4(I,J)=Z5(I,J)
 4022 CONTINUE
       DO 4032 J=1,N
       DO 4032 I=1,M
       Z3(I,J)=-1./Z4(I,J)
 4032 CONTINUE
      ENDIF

C  -- Inverse Theodosen Transformation

      DO 4040 J=1,N
      DO 4040 I=1,M
      W3=CMPLX(A0,B0)
      DO 4041 JC=1,M3
      W1=CMPLX(A(JC),B(JC))
      W3=W3+W1/Z3(I,J)**(JC)
 4041 CONTINUE
      W4=EXP(W3)
      Z2(I,J)=Z3(I,J)*W4
 4040 CONTINUE

C --- Inverse Karman-Trefftz Airfoil

      DO 4051 J=1,N
      DO 4051 I=1,M
      W1=Z2(I,J)-SA
      W2=Z2(I,J)+SA
      W3=(W1/W2)**KAPAA
      Z1(I,J)=(Z1TA-W3*Z1NA)/(1.-W3)
 4051 CONTINUE

C  -Inverse Karman-Trefftz for slat

      IF(LB2) THEN
      DO 4061 J=1,N
      DO 4061 I=1,M
      W1=Z1(I,J)-SS
      W2=Z1(I,J)+SS
      W3=(W1/W2)**KAPAS
      Z(I,J)=(Z1TS-W3*Z1NS)/(1.-W3)
 4061 CONTINUE
      ELSE
```

```
        DO 4062 J=1,N
        DO 4062 I=1,M
        Z(I,J)=Z1(I,J)
 4062 CONTINUE
      ENDIF
      DO 4070 J=1,N
      DO 4070 I=1,M
      XX(I,J)=REAL(Z(I,J))-XROT
      YY(I,J)=AIMAG(Z(I,J))
 4070 CONTINUE
      print*,' *****    AFTER TRANSFORMATION    ******'
      print*,'         AIRFOIL                SLAT'
      print*, '  I       XA      YA,        XS          YS'
      do 4071 i=1,m
       WRITE(6,102)i,z(i,n),z(i,1)
 4071 continue
  102 format(i4,2f9.4, 3x,2f9.4)
        DO 4065 I=1,M
        ICP=I
        IF(XX(I+1,N).GE.XX(I,N)) GO TO 4066
 4065 CONTINUE
 4066 CONTINUE
      afs=0.
      afa=0.
      do 4091 i=1,m-1
      yp=.5*(yy(i,n)+yy(i+1,n))
      da=yp*(xx(i+1,n)-xx(i,n))
      afa=afa+da
      yp=.5*(yy(i,1)+yy(i+1,1))
      da=yp*(xx(i+1,1)-xx(i,1))
      afs=afs+da
 4091 continue
       afa=abs(afa)
       afs=abs(afs)
      WRITE(2) NSING,ICP
      WRITE(2) AFA, AFS
      print*,' *** The Area of Airfoil =  ',afa
      print*,' *** The Area of Slat =  ',afs

      WRITE(12,*) MM1,N,AL
      WRITE(12,1225)  ((XX(I,J),I=1,MM1),J=1,N)
      WRITE(12,1225)  ((YY(I,J),I=1,MM1),J=1,N)
 1225 format(10e12.5)

C   CONSTRUCT UC AND VC

      DO 4501 J=1,N
      DO 4501 I=1,M-1
      DZ5DZE=WROT
      dz4dz5=1.
      dz3dz4=1./(z4(i,j)**2)
      IF(LB2) THEN
      W3=0.
      DO 4601 JC=1,M31
      W1=CMPLX(-C(JC),D(JC))
      W2=CMPLX(C(JC),D(JC))
      RZ=R*Z5(I,J)
      ROZ=R/Z5(I,J)
      RZJ=RZ**(JC)
      ROZJ=ROZ**(JC)
      W3=W3+W1*FLOAT(JC)*RZJ/Z5(I,J)
     %      -W2*FLOAT(JC)*ROZJ/Z5(I,J)
 4601 CONTINUE
      DZ4DZ5=Z4(I,J)/Z5(I,J)+Z4(I,J)*W3

      W1=(Z4(I,J)*ZCABS-1.)**2
```

```
      DZ3DZ4=ZCABS*(ZCABS**2-1.)/(ZCBAR*W1)
      ENDIF

      W3=0.
      DO 4701 JC=1,M3
      W1=CMPLX(A(JC),B(JC))
      W3=W3-W1*FLOAT(JC)/Z3(I,J)**(JC+1)
 4701 CONTINUE
      DZ2DZ3=Z2(I,J)/Z3(I,J)+Z2(I,J)*W3

      W1=((Z2(I,J)-SA)/(Z2(I,J)+SA))**(KAPAA-1.)
      W2=(Z1(I,J)-Z1NA)**2/(Z1TA-Z1NA)
      DZ1DZ2=KAPAA*W2*W1*2.*SA/(Z2(I,J)+SA)**2

      dzdz1=1.
      IF(LB2) THEN
      W1=((Z1(I,J)-SS)/(Z1(I,J)+SS))**(KAPAS-1.)
      W2=(Z(I,J)-Z1NS)**2/(Z1TS-Z1NS)
      DZDZ1=KAPAS*W2*W1*2.*SS/(Z1(I,J)+SS)**2
      ENDIF

      DZDZE=DZDZ1*DZ1DZ2*DZ2DZ3*DZ3DZ4*DZ4DZ5*DZ5DZE
      XC=REAL(DZDZE)
      XN=-AIMAG(DZDZE)
      UC(I,J)=XX(I,J)*XN+YY(I,J)*XC
      VC(I,J)=YY(I,J)*XN-XX(I,J)*XC
      HW1=CABS(DZDZE)**2
      XXC(I,J)=XC/HW1
      XXN(I,J)=XN/HW1

C--- Determine XT YT XB YB Arrays

      IF(J.EQ.N) THEN
      W4=CMPLX(-SNP(2,I),CSP(2,I))
      W3=DZDZE*W4
      XT(I)=REAL(W3)
      YT(I)=AIMAG(W3)
      XB(I)=XX(I,J)
      YB(I)=YY(I,J)
      ENDIF

 4501 CONTINUE
      WRITE(2) UC,VC
      WRITE(2) XT,YT,XB,YB
      WRITE(2) XX,YY

      WRITE(3) H
      WRITE(3) XX,YY
      WRITE(3) PHI
      WRITE(3) XXC,XXN
      WRITE(3) XT,YT,XB,YB
      WRITE(3) R1,R2

C  CONSTRUCT ASA BSA CSA DSA AND ASS BSS CSS DSS ARRAYS

      DO 5501 J=1,N,N-1
      DO 5501 I=1,M-1
      DZ5DZE=WROT
      dz4dz5=1.
      dz3dz4=1./(z4(i,j)**2)
      IF(LB2) THEN
      W3=0.
      DO 5601 JC=1,M31
      W1=CMPLX(-C(JC),D(JC))
      W2=CMPLX(C(JC),D(JC))
      RZ=R*Z5(I,J)
```

```
      ROZ=R/Z5(I,J)
      RZJ=RZ**(JC)
      ROZJ=ROZ**(JC)
      W3=W3+W1*FLOAT(JC)*RZJ/Z5(I,J)
     %      -W2*FLOAT(JC)*ROZJ/Z5(I,J)
 5601 CONTINUE
      DZ4DZ5=Z4(I,J)/Z5(I,J)+Z4(I,J)*W3

      W1=(Z4(I,J)*ZCABS-1.)**2
      DZ3DZ4=ZCABS*(ZCABS**2-1.)/(ZCBAR*W1)
      ENDIF

      W3=0.
      DO 5701 JC=1,M3
      W1=CMPLX(A(JC),B(JC))
      W3=W3-W1*FLOAT(JC)/Z3(I,J)**(JC+1)
 5701 CONTINUE
      DZ2DZ3=Z2(I,J)/Z3(I,J)+Z2(I,J)*W3

      W1=((Z2(I,J)-SA)/(Z2(I,J)+SA))**(KAPAA-1.)
      W2=(Z1(I,J)-Z1NA)**2/(Z1TA-Z1NA)
      DZ1DZ2=KAPAA*W2*W1*2.*SA/(Z2(I,J)+SA)**2

      dzdz1=1.
      IF(LB2) THEN
      W1=((Z1(I,J)-SS)/(Z1(I,J)+SS))**(KAPAS-1.)
      W2=(Z(I,J)-Z1NS)**2/(Z1TS-Z1NS)
      DZDZ1=KAPAS*W2*W1*2.*SS/(Z1(I,J)+SS)**2
      ENDIF

      DZDZE=DZDZ1*DZ1DZ2*DZ2DZ3*DZ3DZ4*DZ4DZ5*DZ5DZE
      XC=REAL(DZDZE)
      XN=-AIMAG(DZDZE)
      UC(I,J)=XX(I,J)*XN+YY(I,J)*XC
      VC(I,J)=YY(I,J)*XN-XX(I,J)*XC
 5501 CONTINUE
      DO 5160 K=1,KFC
      ASA(K)=0.
      BSA(K)=0.
      CSA(K)=0.
      DSA(K)=0.
      ASS(K)=0.
      BSS(K)=0.
      CSS(K)=0.
      DSS(K)=0.
 5160 CONTINUE
      IF(LB2) THEN
      J=1
       DO 5161 I=1,M-1
      COEF(I,1)=-(UC(I,J)*CSP(2,I)+VC(I,J)*SNP(2,I))
      COEF(I,2)=-(VC(I,J)*CSP(2,I)-UC(I,J)*SNP(2,I))
 5161 CONTINUE
      COEF(M,1)=COEF(1,1)
      COEF(M,2)=COEF(1,2)
      DO 5162 I=1,M-1
      IP1=I+1
      DDPH=DPH(I)
      FA=.5*(COEF(I,1)+COEF(IP1,1))
      WAA=FA*DDPH
      ASS(1)=ASS(1)+WAA/PI
      DO 5163 K=2,KFC
         FAC1=1./FLOAT(K-1)
      WAA= FA*FAC1*(SNP(K,IP1)-SNP(K,I))
      WBB=-FA*FAC1*(CSP(K,IP1)-CSP(K,I))
      ASS(K)=ASS(K)+WAA/PI
      BSS(K)=BSS(K)+WBB/PI
```

```
 5163 CONTINUE
      FA=.5*(COEF(I,2)+COEF(IP1,2))
      WAA=FA*DDPH
      CSS(1)=CSS(1)+WAA/PI
      DO 5164 K=2,KFC
         FAC1=1./FLOAT(K-1)
      WAA= FA*FAC1*(SNP(K,IP1)-SNP(K,I))
      WBB=-FA*FAC1*(CSP(K,IP1)-CSP(K,I))
      CSS(K)=CSS(K)+WAA/PI
      DSS(K)=DSS(K)+WBB/PI
 5164 CONTINUE
 5162 CONTINUE
      ENDIF
      J=N
       DO 5166 I=1,M-1
      COEF(I,1)=-(UC(I,J)*CSP(2,I)+VC(I,J)*SNP(2,I))
      COEF(I,2)=-(VC(I,J)*CSP(2,I)-UC(I,J)*SNP(2,I))
 5166 CONTINUE
      COEF(M,1)=COEF(1,1)
      COEF(M,2)=COEF(1,2)
      DO 5167 I=1,M-1
      IP1=I+1
      DDPH=DPH(I)
      FA=.5*(COEF(I,1)+COEF(IP1,1))
      WAA=FA*DDPH
      ASA(1)=ASA(1)+WAA/PI
      DO 5168 K=2,KFC
         FAC1=1./FLOAT(K-1)
      WAA= FA*FAC1*(SNP(K,IP1)-SNP(K,I))
      WBB=-FA*FAC1*(CSP(K,IP1)-CSP(K,I))
      ASA(K)=ASA(K)+WAA/PI
      BSA(K)=BSA(K)+WBB/PI
 5168 CONTINUE
      FA=.5*(COEF(I,2)+COEF(IP1,2))
      WAA=FA*DDPH
      CSA(1)=CSA(1)+WAA/PI
      DO 5169 K=2,KFC
         FAC1=1./FLOAT(K-1)
      WAA= FA*FAC1*(SNP(K,IP1)-SNP(K,I))
      WBB=-FA*FAC1*(CSP(K,IP1)-CSP(K,I))
      CSA(K)=CSA(K)+WAA/PI
      DSA(K)=DSA(K)+WBB/PI
 5169 CONTINUE
 5167 CONTINUE
c        PRINT*,' ASS(1)  CSS(1)  ASA(1)  CSA(1)  ',
c     %           ASS(1),CSS(1),ASA(1),CSA(1)
      WRITE(2) ASS,BSS,CSS,DSS,ASA,BSA,CSA,DSA

      STOP
      END

C   **************** THDSN *********************

      SUBROUTINE THDSN(ALNRI,THEIN,MA,THET)

C  Theodosen transformation which transforms the near circle of
C  the airfoil to a circle

      PARAMETER (II=160,JJ=60,IIN=161,IC=120)
      COMPLEX W1,W2,WW
      COMPLEX YC,Z3
      COMMON/COEF/M3,AO,BO,A(IC),B(IC)
      DIMENSION PHI(II),TH(II)
      DIMENSION YC(II),Z3(II)
      DIMENSION X(II),Y(II),ALNR(II)
      DIMENSION ALNRI(II),THEIN(II),FP2(II)
```

```
      DIMENSION SGMA(11)
      DATA P/1./
      DATA URF,EPS/.06,.000001/
      PI=3.1415926535898
      CALL TNSPL(P,1,MA,THEIN,ALNRI,FP2)
      M32=M3*2
      DPHI=PI/FLOAT(M3)
      PHI(1)=0.
      DO 101 I=2,M32
      PHI(I)=PHI(I-1)+DPHI
  101 CONTINUE

      DO 1 I=1,M3
      TTH=PHI(I+1)
      SGMA(I)=SIN(TTH)/TTH
    1 CONTINUE

      B(M3)=0.

C ---  initialize A(J) and B(J)

      AO=0.
      DO 201 J=1,M3
      A(J)=0.
      B(J)=0.
  201 CONTINUE

c      print*,' ------information of Theodosen convergence '
      ITER=0
 1001 ITER=ITER+1
      DO 305 I=1,M32
      WA=BO
      DO 306 K=1,M3
      TTH=FLOAT(K)*PHI(I)
      WA=WA+B(K)*COS(TTH)-A(K)*SIN(TTH)
  306 CONTINUE
      TH(I)=PHI(I)+WA
  305 CONTINUE
      DO 311 K=1,M32
      IF(TH(K).LT.0.) TH(K)=TH(K)+2.*PI
      IF(TH(K).GT.2.*PI) TH(K)=TH(K)-2.*PI
  311 CONTINUE

      DO 401 K=1,M32
      ALNR(K)=FTNSPL(P,1,TH(K),THEIN,ALNRI,FP2)
  401 CONTINUE

      WA=0.
      DO 402 I=1,M32
      WA=WA+ALNR(I)
  402 CONTINUE
      ERRO=WA/FLOAT(M32)-AO
      AO=AO+ERRO*URF
      WAER=ERRO*ERRO
      DO 403 K=1,M3
      WA=0.
      WB=0.
      DO 404 I=1,M32
      TTH=FLOAT(K)*PHI(I)
      WA=WA+ALNR(I)*COS(TTH)
      WB=WB+ALNR(I)*SIN(TTH)
  404 CONTINUE
      ERRA=WA/FLOAT(M3)-A(K)
      ERRB=WB/FLOAT(M3)-B(K)
      A(K)=A(K)+URF*ERRA
      B(K)=B(K)+URF*ERRB
```

```
      WAER=WAER+ERRA*ERRA+ERRB*ERRB
  403 CONTINUE
      BO=THET
      DO 501 K=1,M3
      BO=BO-B(K)
  501 CONTINUE
      WA=WA/FLOAT(M32)
      IF(WAER.GT.EPS.AND.ITER.LE.300) GO TO 1001

      DO 2 I=1,M3
      A(I)=A(I)*SGMA(I)
      B(I)=B(I)*SGMA(I)
    2 CONTINUE
c      PRINT*,' AO = ',AO,' BO= ',BO
      RETURN
      END

C  ***************** TNSPL ************************

      SUBROUTINE TNSPL(P,IN1,IN2,X,F,FP2)

C  Find f" at each data point for tension spline fit

      PARAMETER (II=160,JJ=60,IIN=200,IC=120)
      DIMENSION X(II),F(II),FP2(II),B(II)
      DIMENSION A(II,II),IPVT(II)
      N=IN2-IN1
      P2=P*P
      DO 1 I=2,N+1
      I1=I+IN1-1
      DI1=X(I1)-X(I1-1)
      DI=X(I1+1)-X(I1)
      B(I-1)=P2*((F(I1+1)-F(I1))/DI-
     %          (F(I1)-F(I1-1))/DI1)
    1 CONTINUE
      DO 2001 I=1,N
      DO 2001 J=1,N
      A(I,J)=0.
 2001 CONTINUE

      DO 2 I=2,N-1
      I1=I+IN1
      DI1=X(I1)-X(I1-1)
      DI=X(I1+1)-X(I1)
      WW=P*DI1
      IF(WW.GT.30.) GO TO 21
      W1=EXP(WW)
      W2=1./W1
      SINH1=.5*(W1-W2)
      TANH1=(W1-W2)/(W1+W2)
      W5=P/SINH1
      W6=P/TANH1
      GO TO 22
   21 W5=0.
      W6=P
   22 CONTINUE
      WW=P*DI
      IF(WW.GT.30.) GO TO 11
      W1=EXP(WW)
      W2=1./W1
      SINH2=.5*(W1-W2)
      TANH2=(W1-W2)/(W1+W2)
      W3=P/SINH2
      W4=P/TANH2
      GO TO 12
   11 W3=0.
```

```
      W4=P
   12 CONTINUE
      A(I,I-1)=1./DI1-W5
      A(I,I)=W6-1./DI1+W4-1./DI
      A(I,I+1)=1./DI-W3
    2 CONTINUE

C ---  I=1

      I=1
      DI1=X(2)-X(1)
      DI=X(3)-X(2)
      WW=P*DI1
      IF(WW.GT.30.)  GO TO 121
      W1=EXP(WW)
      W2=1./W1
      SINH1=.5*(W1-W2)
      TANH1=(W1-W2)/(W1+W2)
      W5=P/SINH1
      W6=P/TANH1
      GO TO 122
  121 W5=0.
      W6=P
  122 CONTINUE
      WW=P*DI
      IF(WW.GT.30.)  GO TO 111
      W1=EXP(WW)
      W2=1./W1
      SINH2=.5*(W1-W2)
      TANH2=(W1-W2)/(W1+W2)
      W3=P/SINH2
      W4=P/TANH2
      GO TO 112
  111 W3=0.
      W4=P
  112 CONTINUE
      A(I,N)=1./DI1-W5
      A(I,I)=W6-1./DI1+W4-1./DI
      A(I,I+1)=1./DI-W3

C --  I=N

      I=N
      DI1=X(N+1)-X(N-1)
      DI=X(N+2)-X(N+1)
      WW=P*DI1
      IF(WW.GT.30.)  GO TO 221
      W1=EXP(WW)
      W2=1./W1
      SINH1=.5*(W1-W2)
      TANH1=(W1-W2)/(W1+W2)
      W5=P/SINH1
      W6=P/TANH1
      GO TO 222
  221 W5=0.
      W6=P
  222 CONTINUE
      WW=P*DI
      IF(WW.GT.30.)  GO TO 211
      W1=EXP(WW)
      W2=1./W1
      SINH2=.5*(W1-W2)
      TANH2=(W1-W2)/(W1+W2)
      W3=P/SINH2
      W4=P/TANH2
      GO TO 212
```

```
  211 W3=0.
      W4=P
  212 CONTINUE
      A(I,I-1)=1./DI1-W5
      A(I,I)=W6-1./DI1+W4-1./DI
      A(I,1)=1./DI-W3

      CALL SGEFA(A,II,N,IPVT,INFO)
      CALL SGESL(A,II,N,IPVT,B,0)
      DO 3 I=1,N
      I1=I+IN1
      FP2(I1)=B(I)
    3 CONTINUE
      FP2(1)=B(N)
      FP2(IN2+1)=B(1)
      RETURN
      END

C  ******************** FTNSPL  ***********************

      FUNCTION FTNSPL(P,IN1,XBAR,XI,F,FP2)

C  Calculate function value FTNSPL knowning F(XI) and XBAR
C  FP2 is calculated by "TNSPL"

      PARAMETER (II=160,JJ=60,IIN=200,IC=120)
      DIMENSION XI(II),FP2(II),F(II)
      I=IN1-1
   19 I=I+1
      DX=XBAR-XI(I+1)
      IF(DX) 30,30,19
   30 DX1=XBAR-XI(I)
      DX2=XI(I+1)-XBAR
      DXX=XI(I+1)-XI(I)
      P2=P*P
      PR=1./P2
      WW=P*DXX
      IF(WW.GT.30.) GO TO 11
      W1=EXP(WW)
      W2=1./W1
      SINGH=(W1-W2)*.5
      WW=P*DX2
      W1=EXP(WW)
      W2=1./W1
      SINGH2=(W1-W2)*.5
      WW=P*DX1
      W1=EXP(WW)
      W2=1./W1
      SINGH1=(W1-W2)*.5
      W3=SINGH2/SINGH
      W4=SINGH1/SINGH
      GO TO 12
   11 W1=P*DX1
      IF(W1.GT.30.) GO TO 31
      W3=1./EXP(W1)
      GO TO 32
   31 W3=0.
   32 CONTINUE
      W1=P*DX2
      IF(W1.GT.30.) GO TO 33
      W4=1./EXP(W1)
      GO TO 12
   33 W4=0.
   12 CONTINUE
      FTNSPL=PR*W3*FP2(I)
     %       +PR*W4*FP2(I+1)
```

```
     %        +(F(I)-PR*FP2(I))*DX2/DXX
     %        +(F(I+1)-PR*FP2(I+1))*DX1/DXX
      RETURN
      END

C    ********************* NEWTON *************************

      SUBROUTINE NEWTON(Z2,Z3)

C    find Z3 from Z2 which are related by Theodosen transformation
C    using newtons iteration technique

      PARAMETER (II=160,JJ=60,IIN=200,IC=120)
      COMPLEX Z2,Z3,ZZ,WA,WB,WC,W1,F,FP
      COMMON/COEF/M3,A0,B0,A(IC),B(IC)
      DATA EPS/.0001/
      ZZ=Z2
      ITER=0
 1001 ITER=ITER+1
      WA=0.
      WB=CMPLX(A0,B0)
      DO 11 J=1,M3
      W1=CMPLX(A(J),B(J))
      WA=WA+W1*FLOAT(J)/ZZ**(J+1)
      WB=WB+W1/ZZ**(J)
   11 CONTINUE
      WC=EXP(WB)
      FP=-WC+ZZ*WC*WA
      F=Z2-ZZ*WC
      Z3=ZZ-F/FP
      ERR=CABS(Z3-ZZ)
      ZZ=Z3
      IF(ERR.GT.EPS.AND.ITER.LT.30) GO TO 1001
    1 CONTINUE
      RETURN
      END

C   *************** FIND ****************************

      SUBROUTINE FIND(ZC,MS,UR)

C   find Z4C for bilinear transformation

      PARAMETER (II=160,JJ=60,IIN=200,IC=120)
      COMPLEX W1,Z3S,Z4S,ZCNEW
      COMPLEX ZC,Z4C,ZCBAR
      COMMON/ZZ/Z3S(II),Z4S(II)
      DATA EPS/0.000001/

      WA=0.
      IND=2
      DO 1 I=2,MS
      WB=CABS(Z3S(1)-Z3S(I))
      IF(WB.GE.WA) THEN
      WA=WB
      IND=I
      ENDIF
    1 CONTINUE
      ZC=.5*(Z3S(1)+Z3S(IND))
c      print*,' --------information of FIND Z4C iteration convergence'
      ITER=0
 1001 ITER=ITER+1
      WR=REAL(ZC)
      WI=AIMAG(ZC)
      ZCBAR=CMPLX(WR,-WI)
      ZCABS=CABS(ZC)
```

```
c       print*,' iter= ',iter

      DO 501 I=1,MS-1
      W1=(Z3S(I)*ZCBAR-ZCABS*ZCABS)/(Z3S(I)*ZCBAR-1.)
      Z4S(I)=W1/ZCABS
  501 CONTINUE
      Z4S(MS)=Z4S(1)

      WA=0.
      IND=2
      DO 2 I=2,MS
      WB=CABS(Z4S(1)-Z4S(I))
      IF(WB.GE.WA) THEN
      WA=WB
      IND=I
      ENDIF
    2 CONTINUE
      Z4C=.5*(Z4S(1)+Z4S(IND))
      W1=(ZCABS-Z4C)/(-Z4C*ZCABS+1.)
      ZCNEW=ZCABS*W1/ZCBAR
      ZC=ZC+UR*(ZCNEW-ZC)
      IF(CABS(Z4C).GT.EPS.AND.ITER.LE.30) GO TO 1001
      RETURN
      END

C   **************** GARCK **********************

      SUBROUTINE GARCK(ALNRI,THEIN,MS)

C   Garrick transformation transforms inner near circle (of slat)
C   to a circle at radius R

      PARAMETER (II=160,JJ=60,IIN=200,IC=120)
      COMPLEX W1,W2,WW
      COMPLEX YC,Z3
      COMMON/COEF1/M3,C(IC),D(IC)
      COMMON/RSS/R
      DIMENSION PHI(II),TH(II)
      DIMENSION YC(II),Z3(II)
      DIMENSION X(II),Y(II),ALNR(II)
      DIMENSION ALNRI(II),THEIN(II),FP2(II)
      DIMENSION A(IC),B(IC)
      DIMENSION SGMA(II)
      DATA P/1./
      DATA URF,EPS/.1,.00001/
      DATA R/1./
      PI=3.1415926535898
      CALL TNSPL(P,1,MS,THEIN,ALNRI,FP2)
      M32=M3*2
      DPHI=PI/FLOAT(M3)
      PHI(1)=0.
      DO 101 I=2,M32
      PHI(I)=PHI(I-1)+DPHI
  101 CONTINUE
      DO 1 I=1,M3
      TTH=PHI(I+1)
      SGMA(I)=SIN(TTH)/TTH
       A(I)=0.
      B(I)=0.
    1 CONTINUE

C ---  initialize C(J) and D(J)

      DO 201 J=1,M3
      C(J)=0.
```

```
      D(J)=0.
  201 CONTINUE

c      print*,' ----------information of GARRICK convergence '

      ITER=0
 1001 ITER=ITER+1
      DO 305 K=1,M32
      WA=0.
      DO 303 J=1,M3
      R2J=R**(2*J)
      TTH=FLOAT(J)*PHI(K)
      WA=WA+(D(J)*R2J+D(J))*COS(TTH)-(C(J)*R2J+C(J))*SIN(TTH)
  303 CONTINUE
      TH(K)=PHI(K)+WA
  305 CONTINUE
      DO 311 K=1,M32
      IF(TH(K).LT.0.) TH(K)=TH(K)+2.*PI
      IF(TH(K).GT.2.*PI) TH(K)=TH(K)-2.*PI
  311 CONTINUE

      DO 401 K=1,M32
      ALNR(K)=FTNSPL(P,1,TH(K),THEIN,ALNRI,FP2)
  401 CONTINUE

      WA=0.
      DO 409 K=1,M32
      WA=WA+ALNR(K)
  409 CONTINUE
      WA=WA/FLOAT(M32)
      ALNRR=WA
      RNEW=EXP(ALNRR)
      ERROR=ABS(RNEW-R)
      R=RNEW
      DO 403 J=1,M3+1
      W2=0.
      DO 404 K=1,M32
      W1=CMPLX(0.,-FLOAT(J-1)*PHI(K))
      W2=W2+(ALNR(K)-ALNRR)*EXP(W1)
  404 CONTINUE
      YC(J)=W2/FLOAT(M32)
  403 CONTINUE
      DO 405 J=1,M3-1
      ERRA=2.*REAL(YC(J+1))-A(J)
      ERRB=-2.*AIMAG(YC(J+1))-B(J)
      A(J)=A(J)+URF*ERRA
      B(J)=B(J)+URF*ERRB
      R2J=R**(2*J)
      C(J)=A(J)/(1.-R2J)
      D(J)=B(J)/(1.-R2J)
  405 CONTINUE
      ERRA =REAL(YC(M3+1))-A(M3)
      A(M3)=A(M3)+URF*ERRA
      R2J=R**(2*M3)
      C(M3)=A(M3)/(1.-R2J)
      IF(ERROR.GT.EPS.AND.ITER.LE.80) GO TO 1001

C  ----- Smooth C and D ----

      DO 2 I=1,M3
      C(I)=C(I)*SGMA(I)
      D(I)=D(I)*SGMA(I)
    2 CONTINUE
      RETURN
      END
```

```
C    ********************* NEWTON1 ************************

      SUBROUTINE NEWTON1(Z2,Z3)

C    find Z3 from Z2 which are related by Garrick transformation
C    using newtons iteration technique

      PARAMETER (II=160,JJ=60,IIN=200,IC=120)
      COMPLEX Z2,Z3,ZZ,WA,WB,WC,W1,W2,F,FP
      COMPLEX RZ,ROZ,RZJ,ROZJ
      COMMON/COEF1/M3,C(IC),D(IC)
      COMMON/RSS/R
      DATA EPS/.0001/
      ZZ=Z2
      ITER=0
 1001 ITER=ITER+1
      WA=0.
      WB=0.
      DO 11 J=1,M3
      W1=CMPLX(-C(J),D(J))
      W2=CMPLX(C(J),D(J))
      RZ=R*ZZ
      ROZ=R/ZZ
      RZJ=RZ**(J)
      ROZJ=ROZ**(J)
      WA=WA+W1*FLOAT(J)*RZJ/ZZ
     %      -W2*FLOAT(J)*ROZJ/ZZ
      WB=WB+W1*RZJ+W2*ROZJ
   11 CONTINUE
      WC=EXP(WB)
      FP=-WC-ZZ*WC*WA
      F=Z2-ZZ*WC
      Z3=ZZ-F/FP
      ERR=CABS(Z3-ZZ)
      ZZ=Z3
C     PRINT*,' ITER ERR = ',ITER,ERR
      IF(ERR.GT.EPS.AND.ITER.LT.80) GO TO 1001
    1 CONTINUE
      RETURN
      END
```

```
      PROGRAM ZONST

C   ************************************************************
C.. Add BL features and no SMOOTH of the Fourier Coeffs.
C.. Separate the Boundary Vorticity by a Potential and a domain
C   vortical parts. The convergence of Surface vorticity is only
C   done on the part influenced by the domain vortial part.
C...  The Fourier Coeffs are up to 240 harmonics (AA(j,240))
C
C    2-D INCOMPRESSIBLE NAVIER-STOKES SOLVER FOR TWO-ELEMENT
C    LAST REVISION:  9-28-94
C
C    AUTHOR : C.M. WANG
C             GEORGIA INSTITUTE OF TECHNOLOGY
C
C    TAPE2   : OUTPUT FROM GEOM
C    TAPE5   : GENERAL INPUT
C    TAPE6   : GENERAL OUTPUT
C    TAPE7   : INPUT FROM PREVIOUS RUN
C    TAPE8   : OUTPUT FOR NEXT RUN
C
C   ************************************************************

      PARAMETER (IDIM=160,JDIM=60)
      PARAMETER (IDP1=161,IDP2=162)
      PARAMETER (KFC1=241,KFC2=242)
      DIMENSION WOW(IDP1,JDIM),POP(IDP1,JDIM),WB(IDP1)
      DIMENSION XT(IDIM),YT(IDIM),XB(IDIM),YB(IDIM)
      DIMENSION WIDN(600),CL(600),CD(600),CM(600)
      DIMENSION ASS(KFC1),BSS(KFC1),CSS(KFC1),DSS(KFC1)
      DIMENSION ASA(KFC1),BSA(KFC1),CSA(KFC1),DSA(KFC1)
      common/ga0/AGA0_S(KFC1),AGA0_A(KFC1),BGA0_S(KFC1),BGA0_A(KFC1)
      COMMON/ga1/AGA1_S(KFC1),AGA1_A(KFC1),BGA1_S(KFC1),BGA1_A(KFC1)
      common/znss/ib1s,iv1s,ib2s,iv2s
      common/znsa/ib1a,iv1a,ib2a,iv2a
      common/tur1/ustrs(idim),ustra(idim),eddy(idim,jdim)
      common/ikp/kps(idim),kpa(idim)
      common/tmark/is1,is2,ia1,ia2
      COMMON/CORD/XX(IDP1,JDIM),YY(IDP1,JDIM)
      COMMON/AOF/SINALP,COSALP
      COMMON/ABCDS/ASR(KFC1),BSR(KFC1),CSR(KFC1),DSR(KFC1)
      COMMON/ABCDA/AS1(KFC1),BS1(KFC1),CS1(KFC1),DS1(KFC1)
      COMMON/VEH/UC(IDIM,JDIM),VC(IDIM,JDIM)
      COMMON/DIVD/NSING,AFA,AFS
      COMMON/DANG/DPH(IDP1)
      COMMON/ANG/PHI(IDP1),TH(IDP1)
      COMMON/VELS/UA(IDIM,JDIM),UB(IDIM,JDIM)
      COMMON/IO/LOOP,NT,NTMAX,NTOUT
      COMMON/VLB/VORLS,NRS,VORLA,NRA
      COMMON/RGRD/R1(JDIM),R2(JDIM)
      COMMON/SMT/SGMA(KFC1)
      COMMON/WFC/AA(JDIM,KFC1),BB(JDIM,KFC1)
      COMMON/VFC/A(JDIM,KFC1),B(JDIM,KFC1),C(JDIM,KFC1),D(JDIM,KFC1)
      COMMON/COR/RP(JDIM,KFC2),RL(JDIM)
      COMMON/DELTA/DT
      COMMON/UNF/URR,DFMX,NCC
      COMMON/VTEXT/KINS(IDIM),KENS(IDIM),KINA(IDIM),KENA(IDIM)
      COMMON/TRIGI/CSP(KFC1,IDP1),SNP(KFC1,IDP1)
      COMMON/SCALE/H(IDIM,JDIM)
      COMMON/VV/VOR(IDIM,JDIM),VOROLD(IDIM,JDIM)
      COMMON/VEL/U(IDIM,JDIM),V(IDIM,JDIM)
      COMMON/GRD/IM,IM2,KFC,N
      COMMON/COE/VSC,NPL,ICST,OMG
      COMMON/CPC/CPA(IDP1),CPS(IDP1)
      COMMON/VCOEFS/ACR(KFC1),BCR(KFC1),CCR(KFC1),DCR(KFC1)
      COMMON/VCOEFA/AC1(KFC1),BC1(KFC1),CC1(KFC1),DC1(KFC1)
```

```
      COMMON/GACOEF/RA,CGAR(KFC1),CGA1(KFC1)
      COMMON/GAS/GAMAS
      COMMON/MARK/ISS,NS,NA
      COMMON/TYPE/LB2
      COMMON/TTUR/TURB
      LOGICAL ANUMOT,TURB,LB2,POT

C..READ GENERAL INPUTS AND ECHO THEM

      OPEN(UNIT=5,FILE='zonst.in',STATUS='OLD',FORM='FORMATTED')
      READ(5,*) ALPI,ICST,FR,IM,N
      READ(5,*) WMIN,DFMX,DRMX,KMAX,NCC
      READ(5,*) URBS,URBA,URBP,URR
      READ(5,*) NPL,NRS,NRA
      READ(5,*) DTI,DTINC,DTMAX,NTMAX
      READ(5,*) NTPL,NTOUT,NTLO
      READ(5,*) ALPMEAN,ALPAMP
      READ(5,*) RE
      READ(5,*) IS1,IS2,IA1,IA2
      READ(5,*) ANUMOT,TURB
      READ(5,*) KFC
      READ(5,*) LB2
      READ(5,*) POT
      READ(5,*) IB1S,IV1S,IV2S,iB2S
      READ(5,*) IV1A,IB1A,IB2A,IV2A

c     WRITE(6,54) ALPI,ICST,IM,N,WMIN,DFMX,DRMX,KMAX,NCC,
c    1 URP,URR,NPL,NRS,NRA,DTI,DTINC,
c    2 DTMAX,NTMAX,NTPL,NTOUT,NTLO

C..INPUTS FROM GEOM

      READ(2) AL,TH6S,NA,NS
      READ(2) H
      READ(2) CSP,SNP
      READ(2) RP,RL,SGMA
      READ(2) R1,R2
      READ(2) TH,PHI
      READ(2) ACR,BCR,CCR,DCR
      READ(2) AC1,BC1,CC1,DC1
      READ(2) RA,CGAR,CGA1
      READ(2) UA,UB
      READ(2) NSING,ICP
      READ(2) AFA,AFS
      READ(2) UC,VC
      READ(2) XT,YT,XB,YB
      READ(2) XX,YY
      READ(2) ASS,BSS,CSS,DSS,ASA,BSA,CSA,DSA

      Print*,' NSING NA NS =  ',nsing,na,ns

      IM2=IM/2
      PI=3.1415926535898
      VSC=AL/RE
      ALP=ALPI*PI/180.
      print*,' *** START ***'

C..START INITIAL SOLUTION OR READ PREVIOUS ITERATION RESULTS

      SINALP=SIN(ALP)
      COSALP=COS(ALP)
      DDTH=PI/FLOAT(KFC-1)
      DO 3321 K=2,KFC
      THA=FLOAT(K-1)*DDTH
      SGMA(K)=SIN(THA)/THA
 3321 CONTINUE
```

```
      PHIMIN=PHI(1)
      II=1
      DO 99 I=2,IM
      IF(PHI(I).LT.PHIMIN) THEN
      II=I
      PHIMIN=PHI(I)
      ENDIF
   99 CONTINUE
      ISS=II-4
      IF(ISS.LE.0) ISS=ISS+IM
      DO 995 I=II,IM
      PHI(I)=PHI(I)+2.*PI
  995 CONTINUE
      PHI(IM+1)=PHI(1)+2.*PI

C   Construct DPH array

         DO 991 I=2,IM
      DPH(I)=.5*(PHI(I+1)-PHI(I-1))
  991 CONTINUE
      DPH(1)=.5*(PHI(2)+2.*PI-PHI(IM))
      IF(ICST.EQ.0) THEN
        NT=0
        T=0.0
        NLOAD=0
        IF(DTI.EQ.0.) DT=0.001
        GAMAS=0.
        VORLS=0.0
        VORLA=0.0
        DO 105 I=1,IM
          KINS(I)=NSING-1
          KINA(I)=N-NSING-1
          DO 105 J=2,N-2
          VOR(I,J)=0.
  105   CONTINUE

C..POTENTIAL FLOW SOLUTION

      if(pot) then
       if(LB2) then
       i1=1
       xmax=xx(1,1)
       do 978 i=2,im
       if(xx(i,1).ge.xmax) xmax=xx(i,1)
       if(xx(i,1).ge.xmax) i1=i
  978  continue
       gamas=0.
       vorls=0.
       vorla=0.
       iter=0
  976 iter=iter+1
      call wsurf
      w1=0.
      w2=0.
       do 986 k=2,kfc
       w1=w1+aa(n-1,k)*csp(k,1)+bb(n-1,k)*snp(k,1)
       w2=w2+aa(1,k)*csp(k,i1)+bb(1,k)*snp(k,i1)
  986 continue
      vorls1=w2*2.*pi*(r1(2)-r1(1))*r2(1)
      dvor=vorls1-vorls
      vorls=vorls+urbs*(vorls1-vorls)
      vorla1=w1 *2.*pi*(r1(n)-r1(n-1))
      vorla=vorla+urba*(vorla1-vorla)
      gama1=vorls+vorla
      dgamas=gama1-gamas
      gamas=gamas+dgamas*(1.0)
```

```
      if(iter.ge.800)  go to 977
      if(abs(dvor).gt.0.00001) go to 976
  977 continue

       ELSE
      gamas=0.
      vorla=gamas
      iter=0
  971  iter=iter+1
       call wsurf
       wl=0.
       do 981 k=2,kfc
       wl=wl+aa(n-1,k)*csp(k,1)+bb(n-1,k)*snp(k,1)
  981  continue
       gamal=pi*2.*(r1(n)-r1(n-1))*wl
      dgama=gamal-gamas
      gamas=gamas+dgama*(1.0)
       vorla=gamas

        if(abs(dgama).lt.0.0001) go to 972
         if(iter.le.100) go to 971
  972 continue
       endif
        endif

        CALL WSURFO
        DN1=R1(2)-R1(1)
        DN2=R1(N)-R1(N-1)
        AA(1,1)=AGAO_S(1)/DN1
        AA(N-1,1)=AGAO_A(1)/DN2

        DO K=2,KFC
        AA(1,K)=AGAO_S(K)/DN1
        BB(1,K)=BGAO_S(K)/DN1
        AA(N-1,K)=AGAO_A(K)/DN2
        BB(N-1,K)=BGAO_A(K)/DN2
        ENDDO

        DO 106 I=1,IM
        W1=AA(1,1)*.5
        W2=AA(N-1,1)*.5
        DO 107 K=2,KFC
        W1=W1+AA(1,K)*CSP(K,I)+BB(1,K)*SNP(K,I)
        W2=W2+AA(N-1,K)*CSP(K,I)+BB(N-1,K)*SNP(K,I)
  107 CONTINUE
       VOR(I,1)=W1/H(I,1)
       VOR(I,N-1)=W2/H(I,N-1)
  106 CONTINUE

        CALL KMTCS

C  -- Print initial solution

      if(pot) go to 999
      ENDIF

C..READ PREVIOUS RUN AND RESET TIME IF A NEW MOTION IS SPECIFIED

      IF(ICST.GE.1) THEN
      READ(7) NT,N,KINS,KINA,T,DT,VOR,U,V,VORLS,VORLA,GAMAS
      READ(7) AGA1_S,AGA1_A,BGA1_S,BGA1_A
      ENDIF
      IF(DTI.NE.0.) DT=DTI

      IF(ANUMOT) THEN
      NT=0
```

```
      T=0.
      NLOAD=0
      ENDIF

C  //////////TIME STEP LOOP///////////////////
C..START COMPUTATIONS FOR SUBSEQUENT TIME STEPS

      do 1991 i=1,im
      kps(i)=0
      kpa(i)=0
 1991 continue
      do 1992 i=is1,is2
      kps(i)=1
 1992 continue
      do 1993 i=ia1,ia2
      kpa(i)=1
 1993 continue

C ..   (DO LOOP 1001)

      DO 1001 LOOP=1,NTMAX
      TO=T
      NTO=NT
      DTO=DT
      DT=DT+DTINC
      IF (DT.GT.DTMAX) DT=DTMAX
      T=T+DT
      NT=NT+1
      DO 110 J=1,N
      DO 110 I=1,IM
  110 VOROLD(I,J)=VOR(I,J)
      IF(ICST.EQ.4) THEN
      ALP=(ALPMEAN-ALPAMP*COS(FR*T))*PI/180.
      OMG=-ALPAMP*FR*SIN(FR*T)*PI/180.
      OMGD=-ALPAMP*FR*FR*COS(FR*T)*PI/180.
      ENDIF
      SINALP=SIN(ALP)
      COSALP=COS(ALP)
      ALPD=ALP*180./PI

      PRINT*,'    '
      PRINT*,'  ----------------------------------------'
      PRINT*,'   ** NT =',nt,'  T = ',t,'  ALP = ',alpd

      DO 115 K=1,KFC
      ASR(K)=OMG*ASS(K)
      BSR(K)=OMG*BSS(K)
      CSR(K)=OMG*CSS(K)
      DSR(K)=OMG*DSS(K)
      AS1(K)=OMG*ASA(K)
      BS1(K)=OMG*BSA(K)
      CS1(K)=OMG*CSA(K)
      DS1(K)=OMG*DSA(K)
  115 CONTINUE

C..SOLVER FOR KINETICS

      CALL KNTCS(WMIN,URBS,URBA,urbp,KMAX,DRMX,IERR,KKK,ALP,LOOP)

c     IF(IERR.EQ.1) GO TO 5000

C..DETERMINE VOR. EXTENT AT EACH RADIAL LINE IN N-S REGION
C  AND STORE IN ARRAY KEN

      DO 118 I=1,IM
      DO 116 J=2,NSING
```

```
  116 IF(ABS(VOR(I,J)).GT.WMIN) KENS(I)=J
      DO 117 J=N-1,NSING,-1
  117 IF(ABS(VOR(I,J)).GT.WMIN) KENA(I)=N-J
  118 CONTINUE

C..UPDATE ARRAY KIN

      DO 120 I=1,IM
        KINS(I)=KENS(I)+2
        KINA(I)=KENA(I)+2
        IF(KINS(I).GT.NSING) KINS(I)=NSING
        IF(KINA(I).GT.N-NSING) KINA(I)=N-NSING
      IF(.NOT.LB2) KINA(I)=MIN(KINA(I),N-2)
  120 CONTINUE

      IF(LB2) THEN
      DO 121 IW=ISS,ISS+8
      I=IW
      IF(I.GT.IM) I=I-IM
      IF(KINS(I).GT.NS-1) KINS(I)=NS-1
      IF(KINA(I).GT.N-NSING-2) KINA(I)=N-NSING-2
  121 CONTINUE
      ENDIF

C..SOLVER FOR KINEMATICS

      CALL KMTCS

      DO 129 IW=ISS,ISS+8
      I=IW
      IF(I.GT.IM) I=I-IM
      DO 129 J=NS,NS+2
      VOR(I,J)=0.
  129 CONTINUE
      PRINT*,' VORLS VORLA GAMAS = ',VORLS,VORLA,GAMAS
 1001 CONTINUE

C  /////////////////END OF TIME STEP////////////////////////

  999 continue
      WRITE(8) NT,N,KINS,KINA,T,DT,VOR,U,V,VORLS,VORLA,GAMAS
      WRITE(8) AGA1_S,AGA1_A,BGA1_S,BGA1_A

C..CALCULATE STREAM FUNCTION AND VORTICITY AT VELOCITY
C  GRID POINTS AND STORE ON TAPE FOR GENERATING PLOTS.
C  INTEGRATION IS FIRST ORDER TRAPAZOIDAL RULE.

C  - GIVE 'ZERO' VALUE ON R=1

        DO 140 I=1,IM
  140   POP(I,N)=0.

C  - FIND PSI-VALUE ON R=R(1) BY INTEGRATING ALONG A SPECIFIC LINE

      IF(LB2) THEN
          DO 141 J=N-1,1,-1
            R1D=R1(J+1)-R1(J)
            POP(IM2,J)=POP(IM2,J+1)+.5*(U(IM2,J+1)+U(IM2,J))*R1D
  141     CONTINUE
           POP1=POP(IM2,1)

C  -- GIVE VALUE ON R=R(1)

          DO 142 I=1,IM
          POP(I,1)=POP1
  142     CONTINUE
```

```
      ENDIF

C..INTEGRATE TANGENTIAL VELOCITY FOR STREAM FUNCTION AND AVERAGE
C  VORTICITY VALUES FOR VELOCITY GRID

        DO 150 I=1,IM
      IF(LB2) THEN
          DO 149 J=2,NS+1
            R1D=R1(J)-R1(J-1)
            WOW(I,J)=.5*(VOR(I,J)+VOR(I,J-1))
            POP(I,J)=POP(I,J-1)-.5*(U(I,J-1)+U(I,J))*R1D
  149     CONTINUE
          WOW(I,1)=2.*VOR(I,1)-WOW(I,2)
      ENDIF
          DO 148 J=N-1,N-NA,-1
            R1D=R1(J+1)-R1(J)
            WOW(I,J)=.5*(VOR(I,J)+VOR(I,J-1))
            POP(I,J)=POP(I,J+1)+.5*(U(I,J+1)+U(I,J))*R1D
  148  CONTINUE
          WOW(I,N)=2.*VOR(I,N-1)-WOW(I,N-2)
  150   CONTINUE
        WRITE(9,*)  IM,N,NT,T,ALP,RE
        WRITE(9,1225)  ((WOW(I,J),I=1,IM),J=1,N)
        WRITE(9,1225)  ((POP(I,J),I=1,IM),J=1,N)
 1225 format(10e12.5)

      WRITE(6,63)  (VOR(I,1),I=1,IM)
      WRITE(6,64)  (VOR(I,N-1),I=1,IM)

      STOP

   05 FORMAT(' ',//1X,'TIME= ',T24,a8,/1X,
     1        'DATE= ',T24,a10,/1X,
     2        'EXECUTION TIME= ',T18,F10.4,' SECONDS')
   50 FORMAT(//10X,'********ABNORMAL EXIT**********')
   51 FORMAT(//20X,'*AIRFOIL MOTION HAS BEEN CHANGED, TIME RESET*',//)
   52 FORMAT(/10X,'*NO CONVER. SOLU. WITHIN',I5,3X,'ITERATIONS*',//10X,
     1 'FOLLOWING VALUES ARE FROM LAST COMPLETED TIME STEP',//)
   53 FORMAT(//,30X,'$$ NEXT RUN WITH NSTART =',I6,'  $$')
   54 FORMAT(//3X,'INPUT FILE TO ZONST',//1X,
     1  'ALPI,ICST,IM,N = ',T25,F7.4,3I5,/1X,
     2  'WMIN,DFMX,DRMX,KMAX,NCC = ',T25,3F7.4,2I5,/1X,
     3  'URP,URR = ',T25,2F7.4,/1X,
     4  'NPL,NRS,NRA = ',T25,3I5,/1X,
     5  'DTI,DTINC,DTMAX,NTMAX = ',T25,3F7.4,I5,/1X,
     6  'NTPL,NTOUT,NTLO = ',T25,3I5,//)
   57 FORMAT(5X,'KINETICS ITS = ',I4,5X,'TVOR = ',E13.6,5X,'VORLS = ',
     1        E13.6,'VORLA = ',E13.6)
   61 FORMAT(/2X,90(1H-)/2X,'NT = ',I4,2X,'T =',F8.4,2X,'DT = ',F8.6,
     1 2X,'AA = ',F7.4,2X,'OMG = ',F6.4,2X,'OMGD = ',F6.4/2X,90(1H-))
   63 FORMAT(//1H ,'VORTICITY ---SLAT RING---'
     1 ,'CCW FROM TRAILING EDGE:'/,/(8F10.2))
   64 FORMAT(//1H ,'VORTICITY ---AIRFOIL RING---'
     1 ,'CCW FROM TRAILING EDGE:'/,/(8F10.2))
   67 FORMAT(/1H ,'TANGENTIAL VELOCITY ---J=2---'
     1 ,'CCW FROM TRAILING EDGE:'/,/(10E13.6))
   68 FORMAT(/1H ,'VORTICITY EXTENT  :      *** KENS '
     1 ,'***',/,/(1H ,4(2X,10I3)))
   69 FORMAT(/1H ,'VORTICITY EXTENT  :      *** KENA '
     1 ,'***',/,/(1H ,4(2X,10I3)))
   70 FORMAT(/1H ,'TURBULENT REGION  :  ',I3,'- 1   AND  ',I3,'-',I3)
      END

C   *******************  KNTIC  ********************************

      SUBROUTINE KNTCS(WMIN,URBS,URBA,URBP,KMAX,DRMX,IERR,KKK,ALP,LOOP)
```

```
C  --------------------------------
C    KINETICS OF THE PROBLEM
C
C    CALLS      : USTARB  WSURF
C                 EDDYS    FOCFT
C                 VORTY
C
C    CALLED BY : ZONST
C  ----------------------------------------

      PARAMETER (IDIM=160,JDIM=60)
      PARAMETER (IDP1=161,IDP2=162)
      PARAMETER (KFC1=241)
      dimension ICBLS(IDIM),ICBLA(IDIM)
      common/ga0/AGA0_S(KFC1),AGA0_A(KFC1),BGA0_S(KFC1),BGA0_A(KFC1)
      COMMON/ga1/AGA1_S(KFC1),AGA1_A(KFC1),BGA1_S(KFC1),BGA1_A(KFC1)
      common/znss/ib1s,iv1s,ib2s,iv2s
      common/znsa/ib1a,iv1a,ib2a,iv2a
      common/ikp/kps(idim),kpa(idim)
      common/smt/sgma(kfc1)
      COMMON/TRIG1/CSP(KFC1,IDP1),SNP(KFC1,IDP1)
      COMMON/DANG/DPH(IDP1)
      COMMON/ANG/PHI(IDP1),TH(IDP1)
      COMMON/DIVD/NSING
      COMMON/IO/NT,NTMAX,NTOUT
      COMMON/SGA/GA(IDIM,JDIM),GB(IDIM,JDIM),GC(IDIM,JDIM)
      COMMON/RHS/GD(IDIM,JDIM),DIP1(IDIM,JDIM),DIM1(IDIM,JDIM)
      COMMON/WFC/AA(JDIM,KFC1),BB(JDIM,KFC1)
      COMMON/COE/VSC,NPL,ICST,OMG
      COMMON/DELTA/DT
      COMMON/GRD/IM,IM2,KFC,N
      COMMON/SCALE/H(IDIM,JDIM)
      COMMON/VV/VOR(IDIM,JDIM),VOROLD(IDIM,JDIM)
      COMMON/VEL/U(IDIM,JDIM),V(IDIM,JDIM)
      COMMON/RGRD/R1(JDIM),R2(JDIM)
      COMMON/FM/GAM(IDP1,JDIM)
      COMMON/MARK/ISS,NS,NA
      COMMON/VLB/VORLS,NRS,VORLA,NRA
      COMMON/GAS/GAMAS
      COMMON/TUR1/USTRS(IDIM),USTRA(IDIM),EDDY(IDIM,JDIM)
      COMMON/TTUR/TURB
      COMMON/TYPE/LB2
      DIMENSION GAMA_S(KFC1),GAMA_A(KFC1),GAMB_S(KFC1),GAMB_A(KFC1)
      LOGICAL TURB,LB2
      PI=3.1415926535898
      IERR=0
       ALORE=VSC

C  -- Construct an array containing PHI information for later use for
C       interpolation

  199 FORMAT(1X,10F8.4)

      IF(TURB) THEN
      CALL USTARB
      CALL EDDYS
      ENDIF

      do i=1,im
      do j=1,n
      dip1(i,j)=0.
      dim1(i,j)=0.
      enddo
       ICBLS(I)=0
       ICBLA(I)=0
```

```
      enddo

C.. Construct index array for BL zones

      DO i=1,im
      if(i.gt.iv1a.and.i.le.ib1a) icb1a(i)=1
      if(i.ge.ib2a.and.i.lt.iv2a) icb1a(i)=2
      ENDdo

      if(1b2) THEN
      do i=1,im
      if(i.ge.ib1s.and.i.lt.iv1s) icb1s(i)=2
      if(i.gt.iv2s.and.i.le.ib2s) icb1s(i)=1
      enddo

      if(ib2s.gt.im) then
      do i=1,ib2s-im
      icb1s(i)=1
      enddo
      endif

      ENDif

      DO 101 I=1,IM
      IP1=I+1
      IM1=I-1
      IF(I.EQ.IM) IP1=1
      IF(I.EQ.1) IM1=IM
      DPHM1=.5*(DPH(I)+DPH(IM1))
      DPHP1=.5*(DPH(IP1)+DPH(I))
      DPHA=DPH(I)
      WM1=1./(DPHM1*DPHA)
      WP1=1./(DPHP1*DPHA)

C..COMPUTE MATRIX ELEMENTS ONE RADIAL LINE AT A TIME

      DO 100 J=2,N-2

C..TIME AND DIFFUSION TERMS

      JP1=J+1
      W3=WM1/R2(J)
      W4=WP1/R2(J)
      W1=R1(JP1)/((R1(JP1)-R1(J))*(R2(JP1)-R2(J)))
      W2=R1(J)/((R1(JP1)-R1(J))*(R2(J)-R2(J-1)))
      T1=H(I,J)*R2(J)/DT
      if(j.le.nsing.and.icb1s(i).eq.1) w3=0.
      if(j.le.nsing.and.icb1s(i).eq.2) w4=0.
      if(j.gt.nsing.and.icb1a(i).eq.1) w3=0.
      if(j.gt.nsing.and.icb1a(i).eq.2) w4=0.
      GA(I,J)=-ALORE*W1
      GB(I,J)=T1+(W1+W2+W3+W4)*ALORE
      GC(I,J)=-ALORE*W2
      DIP1(I,J)=ALORE*W4
      DIM1(I,J)=ALORE*W3
      GD(I,J)=T1*VOROLD(I,J)

C..DISCRETIZATION OF CONVECTION TERM IN RADIAL DIRECTION

      DRHO=R1(JP1)-R1(J)
      VR=V(I,JP1)
      VL=V(I,J)
      W1=R1(JP1)*VR/DRHO
      W2=R1(J)*VL/DRHO
      IF(VR.LT.0.) GA(I,J)=GA(I,J)+W1
      IF(VR.GE.0.) GB(I,J)=GB(I,J)+W1
```

```
      IF(VL.LT.O.) GB(I,J)=GB(I,J)-W2
      IF(VL.GE.O.) GC(I,J)=GC(I,J)-W2

C..DISCRETIZATION OF CONVECTION TERM IN TANGENTIAL DIRECTION

      DTET=DPHA
      UL=0.25*(U(IM1,JP1)+U(IM1,J)+U(I,JP1)+U(I,J))
      UR=0.25*(U(I,JP1)+U(I,J)+U(IP1,JP1)+U(IP1,J))
      W3=UL/DTET
      W4=UR/DTET
      IF(UL.LT.O.) GB(I,J)=GB(I,J)-W3
      IF(UL.GE.O.) DIM1(I,J)=DIM1(I,J)+W3
      IF(UR.LT.O.) DIP1(I,J)=DIP1(I,J)-W4
      IF(UR.GE.O.) GB(I,J)=GB(I,J)+W4
  100 CONTINUE

      IF(TURB) THEN
      DO 105 J=2,N-1
      JP1=J+1
      W3=WM1/R2(J)
      W4=WP1/R2(J)
      W1=R1(JP1)/((R1(JP1)-R1(J))*(R2(JP1)-R2(J)))
      W2=R1(J)/((R1(JP1)-R1(J))*(R2(J)-R2(J-1)))
      T1=H(I,J)*R2(J)/DT
      if(j.le.nsing.and.icbls(i).eq.1) w3=0.
      if(j.le.nsing.and.icbls(i).eq.2) w4=0.
      if(j.gt.nsing.and.icbla(i).eq.1) w3=0.
      if(j.gt.nsing.and.icbla(i).eq.2) w4=0.
      GA(I,J)=GA(I,J)-W1*EDDY(I,JP1)
      GB(I,J)=GB(I,J)
     %           +(W1+W2+W3+W4)*EDDY(I,J)
      GC(I,J)=GC(I,J)-W2*EDDY(I,J-1)
      DIP1(I,J)=DIP1(I,J)+W4*EDDY(IP1,J)
      DIM1(I,J)=DIM1(I,J)+W3*EDDY(IM1,J)
  105 CONTINUE
      ENDIF

  101 CONTINUE

C..END OF MATRIX CONSTRUCTION LOOP (101)

       CALL WSURFO

        DN1=R1(2)-R1(1)
        DN2=R1(N)-R1(N-1)

       print*,'  The Maximum Surface Vorticity Fourier Coeff.'
       print*,'             AIRFOIL                   SLAT'
       print*,'  ITER    AGA1_A   BGA1_A,      AGA1_S      BGA1_S'

       KKK=0

C..VORTICITY CONVERGENCE LOOP

  500 CONTINUE

c     IF(TURB) THEN
c     CALL WSURF2
c      ENDIF

      KKK=KKK+1

       IF(LB2) THEN

C.. Slat Region
```

```
      ibl=ibls
      if(ib2s.gt.ibls) ibl=ibls+im
      call vortys(ib2s,ibl,1,ics,0,ierr)
      if(ibls.lt.ivls) call vortys(ibls-1,ivls,1,ics,2,ierr)
      if(ib2s.gt.iv2s) call vortys(ib2s+1,iv2s,-1,ics,1,ierr)
      call vortys(ivls,iv2s,1,ics,0,ierr)
      ENDIF

C.. Airfoil Region

      call vortya(ibla,ib2a,1,ics,0,ierr)
      if(ivla.lt.ibla) call vortya(ibla+1,ivla,-1,icb1,1,ierr)
      if(iv2a.gt.ib2a) call vortya(ib2a-1,iv2a,1,icb2,2,ierr)
      call vortya(iv2a,ivla+im,1,ics,0,ierr)

      IF(LB2) THEN
      DO 111 IW=ISS,ISS+8
      I=IW
      IF(I.GT.IM) I=I-IM
      DO 111 J=NS,NS+2
      VOR(I,J)=0.
  111 CONTINUE
      ENDIF

C -- Determine vorticity Fourier coeff. in un-equally spaced PHI-grid

      J1=5
      IF(LB2) J1=2
      do 137 j=1,n
      do 137 k=1,kfc
      aa(j,k)=0.
      bb(j,k)=0.
  137 continue
      do 138 i=1,im
      do 138 j=1,n
      gam(i,j)=0.
  138 continue
      DO 139 J=J1,N-2
      DO 140 I=1,IM
      GAM(I,J)=H(I,J)*VOR(I,J)
  140 CONTINUE
      GAM(IM+1,J)=GAM(1,J)
  139 CONTINUE
  751 format(8f10.3)

      DO 141 J=2,N-2
      DO 144 K=1,KFC
      AA(J,K)=0.
      BB(J,K)=0.
  144 CONTINUE
      DO 143 I=1,IM
      IP1=I+1
      FA=.5*(GAM(IP1,J)+GAM(I,J))
      IF(ABS(FA).LE.0.00001) GO TO 143
      W1=FA*(PHI(IP1)-PHI(I))
      AA(J,1)=AA(J,1)+W1/PI
      DO 142 K=2,KFC
      DSNP=SNP(K,IP1)-SNP(K,I)
      DCSP=CSP(K,IP1)-CSP(K,I)
      AA(J,K)=AA(J,K)+FA*DSNP
      BB(J,K)=BB(J,K)-FA*DCSP
  142 CONTINUE
  143 CONTINUE
      DO 145 K=2,KFC
      FAC1=1./(PI*FLOAT(K-1))
      AA(J,K)=AA(J,K)*FAC1
```

```
      BB(J,K)=BB(J,K)*FAC1
  145 CONTINUE
  141 CONTINUE

C.. SMOOTH ON INTERIOR VORTICITY

c       DO 147 K=2,KFC
c      fac=float(kfc-k+1)/float(kfc)
c       fac=sgma(k)
c       do 147 j=2,n-2
c       AA(J,K)=AA(J,K)*fac
c       BB(J,K)=BB(J,K)*fac
c   147 CONTINUE

      DO K=1,KFC
      GAMA_S(K)=AGA1_S(K)
      GAMB_S(K)=BGA1_S(K)
      GAMA_A(K)=AGA1_A(K)
      GAMB_A(K)=BGA1_A(K)
      ENDDO

      CALL WSURF

C..EVALUATE SURFACE VORTICITY FROM FOURIER COEFFICIENTS,
C  TRANSFORM BACK TO PHYSICAL PLANE AND ITERATE FOR
C  CONVERGENCE USING UNDER-RELAXATION TECHNIQUE

      MAA=0
      MAB=0
      MSA=0
      MSB=0
      WAA=0.
      WAB=0.
      WSA=0.
      WSB=0.

       DMAX=0.
       DMAX1=0.

       IF(LB2) THEN
       DO 150 K=1,KFC
       UR1=URBS
       W1=AGA1_S(K)
       WW=W1*UR1+(1.-UR1)*GAMA_S(K)
       WABS=ABS(WW)
       DD=ABS(WW-GAMA_S(K))
       IF(DD.GE.DMAX1) DMAX1=DD
        IF(WABS.GE.WSA) THEN
          WSA=WABS
          MSA=K
        ENDIF
      AGA1_S(K)=WW
  150 CONTINUE
       DO 151 K=1,KFC
       UR1=URBS
       W1=BGA1_S(K)
       WW=W1*UR1+(1.-UR1)*GAMB_S(K)
       WABS=ABS(WW)
       DD=ABS(WW-GAMB_S(K))
       IF(DD.GE.DMAX1) DMAX1=DD
        IF(WABS.GE.WSB) THEN
          WSB=WABS
          MSB=K
        ENDIF
      BGA1_S(K)=WW
  151  CONTINUE
```

```
      ENDIF

      DO 155 K=1,KFC
      UR1=URBA
      W1=AGA1_A(K)
      WW=W1*UR1+(1.-UR1)*GAMA_A(K)
      WABS=ABS(WW)
      DD=ABS(WW-GAMA_A(K))
      IF(DD.GE.DMAX) DMAX=DD
       IF(WABS.GE.WAA) THEN
         WAA=WABS
         MAA=K
       ENDIF
       AGA1_A(K)=WW
  155 CONTINUE
      DO 156 K=2,KFC
      UR1=URBA
      W1=BGA1_A(K)
      WW=W1*UR1+(1.-UR1)*GAMB_A(K)
      WABS=ABS(WW)
      DD=ABS(WW-GAMB_A(K))
      IF(DD.GE.DMAX) DMAX=DD
       IF(WABS.GE.WAB) THEN
         WAB=WABS
         MAB=K
       ENDIF
       BGA1_A(K)=WW
  156 CONTINUE

      WRITE(6,12) KKK,AGA1_A(MAA),BGA1_A(MAB),
     1                 AGA1_S(MSA),BGA1_S(MSB)

       AA(1,1)=(AGA0_S(1)+AGA1_S(1))/DN1
       AA(N-1,1)=(AGA0_A(1)+AGA1_A(1))/DN2

       DO K=2,KFC
       AA(1,K)=(AGA0_S(K)+AGA1_S(K))/DN1
       BB(1,K)=(BGA0_S(K)+BGA1_S(K))/DN1
       AA(N-1,K)=(AGA0_A(K)+AGA1_A(K))/DN2
       BB(N-1,K)=(BGA0_A(K)+BGA1_A(K))/DN2
       ENDDO
      DO 250 I=1,IM
       IF(LB2) W1=.5*AA(1,1)
      W2=.5*AA(N-1,1)
       DO 249 K=2,KFC
       IF(LB2)
     1 W1=W1+AA(1,K)*CSP(K,I)+BB(1,K)*SNP(K,I)
      W2=W2+AA(N-1,K)*CSP(K,I)+BB(N-1,K)*SNP(K,I)
  249 CONTINUE
       IF(LB2) VOR(I,1)=W1/H(I,1)
       VOR(I,N-1)=W2/H(I,N-1)
  250 CONTINUE

C..ADJUST UNDER-RELAXATION PARAMETER URB.
C  EXIT KNTCS IF CONVERGED, CONTINUE ITERATIONS IF NOT.
C  ABORT IF MAXIMUM ITERATIONS EXCEEDED.

      IF(DMAX.LE.DRMX.AND.DMAX1.LE.DRMX) GO TO 501
      IF(TURB) THEN
      CALL USTARB
       ENDIF
      IF(KKK.LT.KMAX) GO TO 500

C.. Determine Vorticity flow out of two regions and GAMAS

  501 CONTINUE
```

```
      dvorls=0.
      dvorla=0.
      DGAMAS=0.
      DO 125 I=1,IM
      DTET=DPH(I)
      IF(LB2) THEN
      VR=V(I,NRS+1)
      IF(VR.GE.0.) vort=.5*(vor(i,nrs)+VOROLD(I,NRS))
      if(vr.lt.0.) vort=.5*(vor(i,nrs+1)+VOROLD(I,NRS+1))
        VSC1=VSC+EDDY(I,NRS+1)
          VSC2=VSC+EDDY(I,NRS)
       DVORLS=DVORLS+DT*DTET*R1(NRS+1)*(VR*VORT
     1  -(VSC1*.5*(vor(i,nrs+1)+VOROLD(I,NRS+1))
     2    -VSC2*.5*(vor(i,nrs)+VOROLD(I,NRS)))
     2    /   (R2(NRS+1)-R2(NRS)))

      ENDIF
      VR=V(I,N-NRA)
      if(vr.ge.0.) vort=.5*(vor(i,n-nra-1)+VOROLD(I,N-NRA-1))
      if(vr.lt.0.) vort=.5*(vor(i,n-nra)+VOROLD(I,N-NRA))
       VSC1=VSC+EDDY(I,N-NRA)
       VSC2=VSC+EDDY(I,N-NRA-1)
        DVORLA=DVORLA-DT*DTET*R1(N-NRA)*(VR*VORT
     1 -(VSC1*.5*(vor(i,n-nra)+VOROLD(I,N-NRA))
     2    -VSC2*.5*(vor(i,n-nra-1)+VOROLD(I,N-NRA-1)))
     2     / (R2(N-NRA)-R2(N-NRA-1)))
  125 CONTINUE
      IF(LB2) THEN
      w1=0.
      do 132 j=nrs+1,n-nra-1
      dr=r1(j+1)-r1(j)
      do 132 i=1,im
      w1=w1+h(i,j)*(vor(i,j)-vorold(i,j))*r2(j)*
     %            dph(i)*dr
  132 continue
      ELSE
       w1=0.
      do 133 j=5,n-nra-1
       dr=r1(j+1)-r1(j)
       do 133 i=1,im
       w1=w1+h(i,j)*(vor(i,j)-vorold(i,j))*r2(j)*
     *    dph(i)*dr
  133 continue
       ENDIF

C  Calculate GAMAS

      VORLA=VORLA+DVORLA
      IF(LB2) THEN
      dgamas=dvorla+dvorls-w1
      VORLS=VORLS+DVORLS
      GAMAS=GAMAS+DGAMAS
      ELSE
      VORLS=0.
      dgamas=dvorla-w1
      gamas=gamas+dgamas
      ENDIF

C..NO CONVERGENCE OCCURED; RETURN TO MAIN WITH IERR=1

      IERR=1
      RETURN
   10 FORMAT(//,25X,'SURFACE VORTICITY INFORMATION:',20X,
     1  'INTERIOR VORTICITY INFORMATION:'/' ITER',6X,
     2  'REL. MAX.    I    VOR(I,N-1)',9X,'MAX. VAL.    I    URB   ')
```

```
   12 FORMAT(I4,2X,2F10.5,3X,2F10.5)
      END

C  ***************  VORTYS *****************************

      SUBROUTINE VORTYS(IS,IL,INC,IC,ICBL,IERR)

C -- Vorcity of Slat Region


C  ---------------------------------------------
C    CALCULATE VORTICITY BY USING LINE-
C    RELAXATION METHOD ON EACH RADIAL LINE
C
C    CALLS     : TRID
C
C    CALLED BY : KNTCS
C  ---------------------------------------------

      PARAMETER  (IDIM=160,JDIM=60)
      PARAMETER  (KFC1=241)
      common/ikp/kps(idim),kpa(idim)
      COMMON/COE/VSC,NPL,ICST,OMG
      COMMON/GRD/IM,IM2,KFC,N
      COMMON/VTEXT/KINS(IDIM),KENS(IDIM),KINA(IDIM),KENA(IDIM)
      COMMON/UNF/URR,DFMX,NCC
      COMMON/SGA/GA(IDIM,JDIM),GB(IDIM,JDIM),GC(IDIM,JDIM)
      COMMON/RHS/GD(IDIM,JDIM),DIP1(IDIM,JDIM),DIM1(IDIM,JDIM)
      COMMON/VV/VOR(IDIM,JDIM),VOROLD(IDIM,JDIM)
      COMMON/SOLV/SUB(JDIM),DIAG(JDIM),SUP(JDIM),RHS(JDIM)
      COMMON/TTUR/TURB
      COMMON/TYPE/LB2
      LOGICAL TURB,LB2

      IF(IERR.EQ.1) RETURN

      do 100 ic=1,NCC
      wmax=0.
      dmax=0.
      DO 110 II=IS,IL,INC
        I=II
        IP1=I+1
        IM1=I-1
        IF(I.GT.IM)  I=II-IM
        IF(IM1.GT.IM)  IM1=IM1-IM
        IF(IP1.GT.IM)  IP1=IP1-IM

C..CONSTRUCT TRIDIAGONAL MATRIX AND SOLVE

      J1=0
      ks=2

c     if(TURB) THEN
c     KS=3
c     if(kps(i).eq.1) ks=2
c     ENDIF

        DO 120 J=ks,KINS(I)
        J1=J1+1
          SUP(J1)=GA(I,J)
          DIAG(J1)=GB(I,J)
          SUB(J1)=GC(I,J)
          RHS(J1)=GD(I,J)+DIP1(I,J)*VOR(IP1,J)+DIM1(I,J)*VOR(IM1,J)
  120   CONTINUE
        RHS(1)=RHS(1)-SUB(1)*VOR(I,ks-1)
        RHS(J1)=RHS(J1)-SUP(J1)*VOR(I,KINS(I)+1)
```

```
      CALL TRID(J1)

C..UNDER-RELAX THE RESULT OF THE MATRIX SOLUTION

     J1=0
      DO 130 J=ks,KINS(I)
      J1=J1+1
      IF(ICBL.NE.0) THEN
       WW=RHS(J1)
      else
          WW=(1.-URR)*VOR(I,J)+URR*RHS(J1)
          WABS=ABS(WW)
          DD=ABS(WW-VOR(I,J))
          WMAX=AMAX1(WMAX,WABS)
          DMAX=AMAX1(DD,DMAX)
      ENDIF
        VOR(I,J)=WW
 130  CONTINUE
 110 CONTINUE

C..IF BOUNDARY LAYER ZONE, RETURN TO KNTCS AFTER SOLVING EXPLICITLY.
C  ELSE ITERATE FOR CONVERGENCE AND RETURN WHEN CONVERGED OR WHEN
C  MAX NUMBER OF ITERATIONS IS EXCEEDED.

     IF(ICBL.NE.0) RETURN
     IF(DMAX/WMAX.LE.DFMX) GO TO 101
 100 CONTINUE
 101 CONTINUE
c     print*,' ITER for VORTYS = ',ic
     return
     end

C  ***************  VORTYA *****************************

     SUBROUTINE VORTYA(IS,IL,INC,IC,ICBL,IERR)

C -- Vorcity of Airfoil Region

C  ---------------------------------------------
C    CALCULATE VORTICITY BY USING LINE-
C    RELAXATION METHOD ON EACH RADIAL LINE
C
C    CALLS     : TRID
C
C    CALLED BY : KNTCS
C  ---------------------------------------------

     PARAMETER (IDIM=160,JDIM=60)
     PARAMETER (KFC1=241)
     common/ikp/kps(idim),kpa(idim)
     COMMON/COE/VSC,NPL,ICST,OMG
     COMMON/GRD/IM,IM2,KFC,N
     COMMON/VTEXT/KINS(IDIM),KENS(IDIM),KINA(IDIM),KENA(IDIM)
     COMMON/UNF/URR,DFMX,NCC
     COMMON/SGA/GA(IDIM,JDIM),GB(IDIM,JDIM),GC(IDIM,JDIM)
     COMMON/RHS/GD(IDIM,JDIM),DIP1(IDIM,JDIM),DIM1(IDIM,JDIM)
     COMMON/VV/VOR(IDIM,JDIM),VOROLD(IDIM,JDIM)
     COMMON/SOLV/SUB(JDIM),DIAG(JDIM),SUP(JDIM),RHS(JDIM)
     COMMON/TTUR/TURB
     COMMON/TYPE/LB2
     LOGICAL TURB,LB2

     IF(IERR.EQ.1) RETURN

     DO 200 IC=1,ncc
     WMAX=0.
```

```
      DMAX=0.
      DO 210 II=IS,IL,INC
        I=II
        IP1=I+1
        IM1=I-1
        if(i.gt.im) i=i-im
        IF(Ip1.gt.IM) IP1=ip1-im
        IF(Im1.gt.im) IM1=im1-IM

C..CONSTRUCT TRIDIAGONAL MATRIX AND SOLVE

      J1=0
      ks=2

c     IF(TURB) THEN
c     KS=3
c     if(kpa(i).eq.1) ks=2
c     ENDIF

        js=n-kina(i)
        DO 220 J=JS,N-ks
        J1=J1+1
          SUP(J1)=GA(I,J)
          DIAG(J1)=GB(I,J)
          SUB(J1)=GC(I,J)
          RHS(J1)=GD(I,J)+DIP1(I,J)*VOR(IP1,J)+DIM1(I,J)*VOR(IM1,J)
  220   CONTINUE
        RHS(1)=RHS(1)-SUB(1)*VOR(I,JS-1)
        RHS(J1)=RHS(J1)-SUP(J1)*VOR(I,N-ks+1)
        CALL TRID(J1)

C..UNDER-RELAX THE RESULT OF THE MATRIX SOLUTION

      J1=0
        DO 230 J=N-KINA(I),N-ks
          J1=J1+1
         IF(ICBL.NE.0) THEN
          WW=RHS(J1)
          ELSE
            WW=(1.-URR)*VOR(I,J)+URR*RHS(J1)
            WABS=ABS(WW)
            DD=ABS(WW-VOR(I,J))
            WMAX=AMAX1(WMAX,WABS)
            DMAX=AMAX1(DD,DMAX)
        ENDIF
         VOR(I,J)=WW
  230   CONTINUE
  210 CONTINUE

C..IF BOUNDARY LAYER ZONE, RETURN TO KNTCS AFTER SOLVING EXPLICITLY.
C  ELSE ITERATE FOR CONVERGENCE AND RETURN WHEN CONVERGED OR WHEN
C  MAX NUMBER OF ITERATIONS IS EXCEEDED.

       IF(ICBL.NE.0) RETURN
      IF(DMAX/WMAX.LE.DFMX) GO TO 201
  200 CONTINUE
      WRITE(6,10)
      WRITE(6,13) DMAX/WMAX,DFMX
  201 CONTINUE
c      print*,' ITER for VORTYA = ',ic
      RETURN
   10 FORMAT(2X,'NO TRID CONVERGENCE ')
   11 FORMAT(2X,'IS,IL,NCC,ICBL= ',4I7)
   12 FORMAT(2X,'DMAX,WMAX= ',2F15.5)
   13 FORMAT(2X,'DMAX/WMAX,DFMX= ',2F15.5)
      END
```

```
C  *******************  TRID  ************************

      SUBROUTINE TRID(N)

C  --------------------------------------------
C    TRIDIAGONAL MATRIX SOLVER
C
C    CALLS     : NONE
C
C    CALLED BY : VORTY
C  --------------------------------------------

      PARAMETER (JDIM=60)
      COMMON/SOLV/SUB(JDIM),DIAG(JDIM),SUP(JDIM),RHS(JDIM)
      DO 100 J=2,N
      RATIO=-SUB(J)/DIAG(J-1)
      DIAG(J)=DIAG(J)+RATIO*SUP(J-1)
      RHS(J)=RHS(J)+RATIO*RHS(J-1)
  100 CONTINUE
      RHS(N)=RHS(N)/DIAG(N)
      DO 110 J=N-1,1,-1
  110 RHS(J)=(RHS(J)-SUP(J)*RHS(J+1))/DIAG(J)
      RETURN
      END

C  ********************* WSURFO ***************************

      SUBROUTINE WSURFO

C  ----------------------------------------------
C    COMPUTE SURFACE VORTICITY
C
C    CALLS     : NONE
C
C    CALLED BY : KNTCS
C  ----------------------------------------------

      PARAMETER (JDIM=60)
      PARAMETER (KFC1=241,KFC2=242)
      COMMON/RGRD/R1(JDIM),R2(JDIM)
      COMMON/AOF/SINALP,COSALP
      COMMON/COE/VSC,NPL,ICST,OMG
      COMMON/GRD/IM,IM2,KFC,N
      COMMON/COR/RP(JDIM,KFC2),RL(JDIM)
      COMMON/VLB/VORLS,NRS,VORLA,NRA
      COMMON/VCOEFS/ACR(KFC1),BCR(KFC1),CCR(KFC1),DCR(KFC1)
      COMMON/VCOEFA/AC1(KFC1),BC1(KFC1),CC1(KFC1),DC1(KFC1)
      COMMON/ABCDS/ASR(KFC1),BSR(KFC1),CSR(KFC1),DSR(KFC1)
      COMMON/ABCDA/AS1(KFC1),BS1(KFC1),CS1(KFC1),DS1(KFC1)
      COMMON/GACOEF/RA,CGAR(KFC1),CGA1(KFC1)
      COMMON/GAS/GAMAS
      common/divd/nsing,afa,afs
      COMMON/TYPE/LB2
      common/ga0/AGAO_S(KFC1),AGAO_A(KFC1),BGAO_S(KFC1),BGAO_A(KFC1)
      LOGICAL LB2

C..DETERMINE FOURIER COEFFICIENTS OF BOUNDARY VORTICITY,
C  AA(1,K), BB(1,K) FOR K.GE.4

      PI=3.1415926535898
      PAR=GAMAS/PI
      DO 100 K=3,KFC
          KP1=K+1
         RKP1=1./FLOAT(KP1)
          KM1=K-1
```

```
            KM2=K-2
          GNP=2.*(-DCR(K)*COSALP+BCR(K)*SINALP)
     %       -AS1(K)*RP(1,KM2)+ASR(K)+DS1(K)*RP(1,KM2)-DSR(K)
          GN=2.*(-CCR(K)*COSALP+ACR(K)*SINALP)+PAR*CGAR(K)
     %       +BS1(K)*RP(1,KM2)-BSR(K)+CS1(K)*RP(1,KM2)-CSR(K)
          FN=-2.*(-CC1(K)*COSALP+AC1(K)*SINALP)-PAR*CGA1(K)
     %       +CS1(K)-BS1(K)+BSR(K)*RP(1,K)-CSR(K)*RP(1,K)
          FNP=-2.*(-DC1(K)*COSALP+BC1(K)*SINALP)
     %       +DS1(K)+AS1(K)-ASR(K)*RP(1,K)-DSR(K)*RP(1,K)
c          IF(LB2) THEN
            W1=1.-RP(1,KM1)*RP(1,KM1)
            W2=1./(W1)
          BGAO_S(K)=(GNP-FNP*RP(1,KM2))*W2
          BGAO_A(K)=(FNP-GNP*RP(1,K))*W2
          AGAO_S(K)=(GN-FN*RP(1,KM2))*W2
          AGAO_A(K)=(FN-GN*RP(1,K))*W2
c        ELSE
c         w3=1./delsn
c         bb(n-1,k)=fnp*w3
c         aa(n-1,k)=fn*w3
c        ENDIF
  100 CONTINUE
  555 FORMAT(I5,4F10.5)

C..DETERMINE AA(1,K), BB(1,K), FOR K.LE.3

C -- K=1

c        IF(LB2) THEN
        AGAO_S(1)=(-vor1s/pi-csr(1)*rp(1,1))/r2(1)
c        ENDIF
C
        AGAO_A(1)=(-vor1a/pi+cs1(1)-2.*afa*omg/pi)/r2(n-1)

C  -- K=2

        K=2
          GNP=2.*(-DCR(K)*COSALP+BCR(K)*SINALP)
     %       -AS1(K)+ASR(K)+DS1(K)-DSR(K)
          GN=2.*(-CCR(K)*COSALP+ACR(K)*SINALP)+PAR*CGAR(K)
     %       +BS1(K)-BSR(K)+CS1(K)-CSR(K)
          FN=-2.*(-CC1(K)*COSALP+AC1(K)*SINALP)-PAR*CGA1(K)
     %       +CS1(K)-BS1(K)+BSR(K)*RP(1,K)-CSR(K)*RP(1,K)
          FNP=-2.*(-DC1(K)*COSALP+BC1(K)*SINALP)
     %       +DS1(K)+AS1(K)-ASR(K)*RP(1,K)-DSR(K)*RP(1,K)
c          IF(LB2) THEN
            W1=1.-RP(1,1)*RP(1,1)
            W2=1./(W1)
          BGAO_S(K)=(GNP-FNP)*W2
          BGAO_A(K)=(FNP-GNP*RP(1,K))*W2
          AGAO_S(K)=(GN-FN)*W2
          AGAO_A(K)=(FN-GN*RP(1,K))*W2
c        ELSE
c         w3=1./delsn
c         aa(n-1,k)=fn*w3
c         bb(n-1,k)=fnp*w3
c        ENDIF

C..INCLUDE SMOOTHING FUNCTION

c        DO 190 K=2,KFC
cc      fac=float(kfc-k+1)/float(kfc)
c        fac=sgma(k)
c          AA(1,K)=AA(1,K)*fac
c          BB(1,K)=BB(1,K)*fac
c          AA(N-1,K)=AA(N-1,K)*fac
```

```
c         BB(N-1,K)=BB(N-1,K)*fac
c  190 CONTINUE
      RETURN
      END

C  ******************** WSURF ***************************

      SUBROUTINE WSURF

C  ---------------------------------------------
C    COMPUTE SURFACE VORTICITY
C
C    CALLS     : NONE
C
C    CALLED BY : KNTCS
C  ---------------------------------------------

      PARAMETER  (JDIM=60)
      PARAMETER  (KFC1=241,KFC2=242)
      COMMON/ga1/AGA1_S(KFC1),AGA1_A(KFC1),BGA1_S(KFC1),BGA1_A(KFC1)
      COMMON/RGRD/R1(JDIM),R2(JDIM)
      COMMON/WFC/AA(JDIM,KFC1),BB(JDIM,KFC1)
      COMMON/SMT/SGMA(KFC1)
      COMMON/COE/VSC,NPL,ICST,OMG
      COMMON/GRD/IM,IM2,KFC,N
      COMMON/COR/RP(JDIM,KFC2),RL(JDIM)
      COMMON/VLB/VORLS,NRS,VORLA,NRA
      COMMON/GAS/GAMAS
      common/divd/nsing,afa,afs
      COMMON/TYPE/LB2
      LOGICAL LB2

C..DETERMINE FOURIER COEFFICIENTS OF BOUNDARY VORTICITY,
C  AA(1,K), BB(1,K) FOR K.GE.4

      DO 100 K=4,KFC
          KP1=K+1
        RKP1=1./FLOAT(KP1)
          KM1=K-1
          KM2=K-2
          KM3=K-3
        RKM3=1./FLOAT(KM3)
        GNP=0.
        GN=0.
        FN=0.
        FNP=0.
        DO 101 J=2,N-2
          BJ=RP(1,KM2)*RKM3*(1./RP(J,KM3)-1./RP(J+1,KM3))
          GNP=GNP-BB(J,K)*BJ
          GN=GN-AA(J,K)*BJ
          AJ=(RP(J+1,KP1)-RP(J,KP1))*RKP1
          FNP=FNP-BB(J,K)*AJ
          FN=FN-AA(J,K)*AJ
  101   CONTINUE
c       IF(LB2) THEN
          W1=1.-RP(1,KM1)*RP(1,KM1)
          W2=1./(W1)
        BGA1_S(K)=(GNP-FNP*RP(1,KM2))*W2
        BGA1_A(K)=(FNP-GNP*RP(1,K))*W2
        AGA1_S(K)=(GN-FN*RP(1,KM2))*W2
        AGA1_A(K)=(FN-GN*RP(1,K))*W2
c      ELSE
c       w3=1./delsn
c       bb(n-1,k)=fnp*w3
c       aa(n-1,k)=fn*w3
c      ENDIF
```

```
  100 CONTINUE
  555 FORMAT(I5,4F10.5)

C..DETERMINE AA(1,K), BB(1,K), FOR K.LE.3

C -- K=1

c        IF(LB2) THEN
c        A1=.5*(RP(2,2)-RP(1,2))
      FN=0.
      DO 130 J=2,NRS
      AJ=.5*(RP(J+1,2)-RP(J,2))
      FN=FN-AJ*AA(J,1)
  130 CONTINUE
      AGA1_S(1)=(fn)/R2(1)
c        ENDIF

c        ANM1=.5*(RP(N,2)-RP(N-1,2))
      FN=0.
      DO 131 J=N-NRA,N-2
      AJ=.5*(RP(J+1,2)-RP(J,2))
      FN=FN-AJ*AA(J,1)
  131 CONTINUE
      AGA1_A(1)=fn/R2(N-1)

C  -- K=2

      K=2
         KP1=K+1
        RKP1=1./FLOAT(KP1)
        GNP=0.
        GN=0.
        FN=0.
        FNP=0.
        DO 140 J=2,N-2
          BJ=RP(J+1,1)-RP(J,1)
          GNP=GNP-BB(J,K)*BJ
          GN=GN-AA(J,K)*BJ
          AJ=(RP(J+1,KP1)-RP(J,KP1))*RKP1
          FNP=FNP-BB(J,K)*AJ
          FN=FN-AA(J,K)*AJ
  140   CONTINUE
c       IF(LB2) THEN
          W1=1.-RP(1,1)*RP(1,1)
          W2=1./(W1)
        BGA1_S(K)=(GNP-FNP)*W2
        BGA1_A(K)=(FNP-GNP*RP(1,K))*W2
        AGA1_S(K)=(GN-FN)*W2
        AGA1_A(K)=(FN-GN*RP(1,K))*W2
c       ELSE
c        w3=1./delsn
c        aa(n-1,k)=fn*w3
c        bb(n-1,k)=fnp*w3
c       ENDIF

C  -- K=3

      K=3
        KP1=K+1
       RKP1=1./FLOAT(KP1)
        GNP=0.
        GN=0.
        FN=0.
        FNP=0.
        DO 150 J=2,N-2
          BJ=RP(1,1)*(RL(J+1)-RL(J))
```

```
          GNP=GNP-BB(J,K)*BJ
          GN=GN-AA(J,K)*BJ
          AJ=(RP(J+1,KP1)-RP(J,KP1))*RKP1
          FNP=FNP-BB(J,K)*AJ
          FN=FN-AA(J,K)*AJ
  150   CONTINUE
c      IF(LB2) THEN
          W1=1.-RP(1,K-1)*RP(1,K-1)
          W2=1./(W1)
        BGA1_S(K)=(GNP-FNP*RP(1,K-2))*W2
        BGA1_A(K)=(FNP-GNP*RP(1,K))*W2
        AGA1_S(K)=(GN-FN*RP(1,K-2))*W2
        AGA1_A(K)=(FN-GN*RP(1,K))*W2
c      ELSE
c       w3=1./delsn
c       aa(n-1,k)=fn*w3
c       bb(n-1,k)=fnp*w3
c      ENDIF

C..INCLUDE SMOOTHING FUNCTION

c      DO 190 K=2,KFC
cc    fac=float(kfc-k+1)/float(kfc)
c      fac=sgma(k)
c        AA(1,K)=AA(1,K)*fac
c        BB(1,K)=BB(1,K)*fac
c        AA(N-1,K)=AA(N-1,K)*fac
c        BB(N-1,K)=BB(N-1,K)*fac
c  190 CONTINUE
      RETURN
      END

C  **********************  KMTCS  *********************

      SUBROUTINE KMTCS

C  -----------------------------------
C    KINEMATICS OF THE PROBLEM
C
C    CALLS     : NONE
C
C    CALLED BY : ZONST
C  -----------------------------------

      PARAMETER (IDIM=160,JDIM=60)
      PARAMETER (IDP1=161)
      PARAMETER (KFC1=241,KFC2=242)
      common/divd/nsing
      COMMON/AOF/SINALP,COSALP
      COMMON/ABCDS/ASR(KFC1),BSR(KFC1),CSR(KFC1),DSR(KFC1)
      COMMON/ABCDA/AS1(KFC1),BS1(KFC1),CS1(KFC1),DS1(KFC1)
      COMMON/VEH/UC(IDIM,JDIM),VC(IDIM,JDIM)
      COMMON/TRIG1/CSP(KFC1,IDP1),SNP(KFC1,IDP1)
      COMMON/ANG/PHI(IDP1),TH(IDP1)
      COMMON/VELS/UA(IDIM,JDIM),UB(IDIM,JDIM)
      COMMON/COE/VSC,NPL,ICST,OMG
      COMMON/VTEXT/KINS(IDIM),KENS(IDIM),KINA(IDIM),KENA(IDIM)
      COMMON/SMT/SGMA(KFC1)
      COMMON/VFC/A(JDIM,KFC1),B(JDIM,KFC1),C(JDIM,KFC1),D(JDIM,KFC1)
      COMMON/WFC/AA(JDIM,KFC1),BB(JDIM,KFC1)
      COMMON/COR/RP(JDIM,KFC2),RL(JDIM)
      COMMON/VV/VOR(IDIM,JDIM),VOROLD(IDIM,JDIM)
      COMMON/VEL/U(IDIM,JDIM),V(IDIM,JDIM)
      COMMON/GRD/IM,IM2,KFC,N
      COMMON/GACOEF/RA,CGAR(KFC1),CGA1(KFC1)
      COMMON/RGRD/R1(JDIM),R2(JDIM)
```

```
      COMMON/GAS/GAMAS
      DIMENSION cwk1(jdim),cwk2(jdim)
      COMMON/TYPE/LB2
      LOGICAL LB2

      PI=3.1415926535898
      PAR=GAMAS/(2.*PI)
      DO 100 J=2,N-1
      K=2
       W1=RP(1,K)/RP(J,K)
        A(J,K)=.5*( (AS1(K)-DS1(K))
     %             +W1*(ASR(K)+DSR(K)))
        B(J,K)=.5*( (BS1(K)+CS1(K))
     %             +W1*(BSR(K)-CSR(K)))
        C(J,K)=.5*( (BS1(K)+CS1(K))
     %             +W1*(CSR(K)-BSR(K)))
        D(J,K)=.5*( (DS1(K)-AS1(K))
     %             +W1*(ASR(K)+DSR(K)))
      DO 100 K=3,KFC
       W1=RP(1,K)/RP(J,K)
        A(J,K)=.5*( RP(J,K-2)*(AS1(K)-DS1(K))
     %             +W1*(ASR(K)+DSR(K)))
        B(J,K)=.5*( RP(J,K-2)*(BS1(K)+CS1(K))
     %             +W1*(BSR(K)-CSR(K)))
        C(J,K)=.5*( RP(J,K-2)*(BS1(K)+CS1(K))
     %             +W1*(CSR(K)-BSR(K)))
        D(J,K)=.5*( RP(J,K-2)*(DS1(K)-AS1(K))
     %             +W1*(ASR(K)+DSR(K)))
  100 CONTINUE

      PAR=GAMAS/(2.*PI)
      DO 500 J=2,N-1
      JM1=J-1

C..BOUNDARY VELOCITY CONTRIBUTIONS
C..INTEGRATE ALONG RADIAL AXIS FOR AA(J,K), BB(J,K); (K.GE.4)

      DO 120 K=4,KFC
         KP1=K+1
         KM3=K-3
      CW=0.
        CW1=0.
        DW=0.
        DW1=0.
        DO 119 JJ=1,JM1
          W1=RP(JJ+1,KP1)-RP(JJ,KP1)
      CW=AA(JJ,K)*W1+CW
          DW=BB(JJ,K)*W1+DW
  119   CONTINUE
          DO 118 JJ=J,N-1
            W1=1./RP(JJ,KM3)-1./RP(JJ+1,KM3)
            CW1=AA(JJ,K)*W1+CW1
            DW1=BB(JJ,K)*W1+DW1
  118     CONTINUE
        W2=.5/(FLOAT(KP1)*RP(J,K))
        CW2=W2*CW
        DW2=W2*DW
        W3=.5*RP(J,K-2)/FLOAT(KM3)
        CW3=W3*CW1
        DW3=W3*DW1
        C(J,K)=C(J,K)+CW2-CW3
        D(J,K)=D(J,K)+DW2-DW3
        A(J,K)=A(J,K)+DW2+DW3
        B(J,K)=B(J,K)-CW2-CW3
  120 CONTINUE
```

```
C..COMPUTE AA(J,K), BB(J,K); (K.LE.3)

      CW=0.
      CW1=0.
      CW2=0.
      CW3=0.
      CW4=0.
      DW1=0.
      DW2=0.
      DW3=0.
      DW4=0.
      DO 130 JJ=1,JM1
        CW1=CW1+AA(JJ,2)*(RP(JJ+1,3)-RP(JJ,3))
        DW1=DW1+BB(JJ,2)*(RP(JJ+1,3)-RP(JJ,3))
        CW2=CW2+AA(JJ,3)*(RP(JJ+1,4)-RP(JJ,4))
        DW2=DW2+BB(JJ,3)*(RP(JJ+1,4)-RP(JJ,4))
  130 CONTINUE
        DO 140 JJ=J,N-1
          CW3=CW3+AA(JJ,2)*(RP(JJ+1,1)-RP(JJ,1))
          DW3=DW3+BB(JJ,2)*(RP(JJ+1,1)-RP(JJ,1))
          CW4=CW4+AA(JJ,3)*(RL(JJ+1)-RL(JJ))
          DW4=DW4+BB(JJ,3)*(RL(JJ+1)-RL(JJ))
  140   CONTINUE
      CW3=.5*CW3
      DW3=.5*DW3
      W2=1./(6.*RP(J,2))
      CW1=CW1*W2
      DW1=DW1*W2
      C(J,2)=C(J,2)+CW1-CW3
      D(J,2)=D(J,2)+DW1-DW3
      A(J,2)=A(J,2)+DW1+DW3
      B(J,2)=B(J,2)-CW1-CW3
      W2=1./(8.*RP(J,3))
      CW2=CW2*W2
      DW2=DW2*W2
      W2=.5*RP(J,1)
      CW4=CW4*W2
      DW4=DW4*W2
      C(J,3)=C(J,3)+CW2-CW4
      D(J,3)=D(J,3)+DW2-DW4
      A(J,3)=A(J,3)+DW2+DW4
      B(J,3)=B(J,3)-CW2-CW4
  500 CONTINUE
      IF(LB2) THEN
      ns1=nsing
      do 600 j=2,ns1+1
      cw=0.
      do 630 jj=1,j-1
      cw=cw+aa(jj,1)*(rp(jj+1,2)-rp(jj,2))
  630 continue
      cwk1(j)=CSR(1)*RP(1,1)/RP(J,1)+.5*cw/rp(j,1)
  600 continue
      do 700 j=n-1,ns1-1,-1
      cw=0.
      do 730 jj=j,n-1
      cw=cw-aa(jj,1)*(rp(jj+1,2)-rp(jj,2))
  730 continue
      cwk2(j)=CS1(1)/RP(J,1)-PAR*CGA1(1)/RP(J,1)+.5*cw/rp(j,1)
  700 continue
      do 750 j=2,ns1-2
      c(j,1)=cwk1(j)
  750 continue
      do 751 j=n-1,ns1+2,-1
      c(j,1)=cwk2(j)
  751 continue
      do 752 j=ns1-1,ns1+1
```

```
      c(j,1)=.5*(cwk1(j)+cwk2(j))
  752 continue
      ELSE
       do 761 j=n-1,1,-1
       cw=0.
       do 762 jj=j,n-1
       cw=cw-aa(jj,1)*(rp(jj+1,2)-rp(jj,2))
  762 continue
       c(j,1)=cs1(1)/rp(j,1)-par*cga1(1)/rp(j,1)+.5*cw/rp(j,1)
  761 continue
      ENDIF

C..INCLUDE SMOOTHING FUNCTION

c      DO 150 K=2,KFC
c     FAC=FLOAT(KFC-K+1)/FLOAT(KFC)
c      fac=sgma(k)
c      DO 150 J=2,N-1
c       A(J,K)=A(J,K)*FAC
c        B(J,K)=B(J,K)*FAC
c        C(J,K)=C(J,K)*FAC
c        D(J,K)=D(J,K)*FAC
c  150 CONTINUE

C -- Compute velocity at the nodal pointS

      DO 200 J=2,N-1
      DO 201 I=1,IM
      U(I,J)=C(J,1)*.5
      V(I,J)=A(J,1)*.5
        DO 202 K=2,KFC
        U(I,J)=U(I,J)+C(J,K)*CSP(K,I)+D(J,K)*SNP(K,I)
        V(I,J)=V(I,J)+A(J,K)*CSP(K,I)+B(J,K)*SNP(K,I)
  202   CONTINUE
  201 CONTINUE
  200   CONTINUE

C -- ADD THE GAMAS VELOCITY CONTRIBUTIONS

      RA2=RA*RA
      DO 250 J=2,N-1
      R1J2=R1(J)*R1(J)
      DO 250 I=1,IM
      W1=R1(J)-RA*CSP(2,I)
      W2=RA*SNP(2,I)
      WW=R1J2+RA2-2.*RA*R1(J)*CSP(2,I)
      U(I,J)=U(I,J)+PAR*W1/WW
      V(I,J)=V(I,J)-PAR*W2/WW
  250 CONTINUE

C -- ADD THE SINGULAR VELOCITY CONTRIBUTIONS

      DO 300 J=2,N-1
      DO 300 I=1,IM
      U(I,J)=U(I,J)-COSALP*UB(I,J)+SINALP*UA(I,J)
      V(I,J)=V(I,J)+COSALP*UA(I,J)+SINALP*UB(I,J)
  300 CONTINUE

C  TRANSFER TO ROTATING COOR.

      DO 400 I=1,IM
      DO 400 J=2,N-1
      W1=V(I,J)*CSP(2,I)-U(I,J)*SNP(2,I)
      W2=V(I,J)*SNP(2,I)+U(I,J)*CSP(2,I)
      W1=W1+OMG*UC(I,J)
      W2=W2+OMG*VC(I,J)
```

```
      U(I,J)=W2*CSP(2,I)-W1*SNP(2,I)
      V(I,J)=W1*CSP(2,I)+W2*SNP(2,I)
  400 CONTINUE

      RETURN
      END

C ************* USTARB **********************

      SUBROUTINE USTARB

C  -------------------------------------------------------------
C     EVALUATE USTAR ON SOLID BOUNDARIES
C
C    CALLS     : USTRF
C
C    CALLED BY : KNTCS
C--------------------------------------------------------------
      PARAMETER (IDIM=160,JDIM=60)
      PARAMETER (IDP1=161)
      PARAMETER (KFC1=241)
      COMMON/CORD/XX(IDP1,JDIM),YY(IDP1,JDIM)
      COMMON/VV/VOR(IDIM,JDIM),VOROLD(IDIM,JDIM)
      COMMON/TUR1/USTRS(IDIM),USTRA(IDIM),EDDY(IDIM,JDIM)
      COMMON/COE/VSC,NPL,ICST,OMG
      COMMON/GRD/IM,IM2,KFC,N
      COMMON/TYPE/LB2
      LOGICAL LB2

C..EVALUATE USTRS IN SLAT REGION

      OMG2=2.*OMG
      IF(LB2) THEN
      DO 100 I=1,IM
      DX=XX(I,2)-XX(I,1)
      DY=YY(I,2)-YY(I,1)
      YN=SQRT(DX*DX+DY*DY)
      GM=(VOR(I,1)-OMG2)*YN
      AGM=ABS(GM)
      YP=SQRT(AGM*YN/VSC)
       USTRS(I)=AGM/YP
      IF(YP.GT.5.) THEN
       USTRS(I)=USTRF(USTRS(I),AGM,YN,2.5,5.5,YP)
      IF(YP.LT.30.) USTRS(I)=USTRF(USTRS(I),AGM,YN,5.,-3.05,YP)
        ENDIF
  100 CONTINUE
      ENDIF

C..EVALUATE USTRA IN AIRFOIL REGION

      DO 200 I=1,IM
      DX=XX(I,N-1)-XX(I,N)
      DY=YY(I,N-1)-YY(I,N)
      YN=SQRT(DX*DX+DY*DY)
      GM=(VOR(I,N-1)-OMG2)*YN
      AGM=ABS(GM)
      YP=SQRT(AGM*YN/VSC)
       USTRA(I)=AGM/YP
      IF(YP.GT.5.) THEN
            USTRA(I)=USTRF(USTRA(I),AGM,YN,2.5,5.5,YP)
      IF(YP.LT.30.) USTRA(I)=USTRF(USTRA(I),AGM,YN,5.,-3.05,YP)
      ENDIF
  200 CONTINUE
      RETURN
      END
```

```
C *************** USTRF ****************************************

      FUNCTION USTRF(USTR,UV,YN,A,C,YP)
C  ----------------------------------------
C    CALLS     : NONE
C
C    CALLED BY : WSURFT
C  ----------------------------------------
      COMMON/COE/VSC,NPL,ICST,OMG
      W1=YN/VSC
      DO 100 I=1,20
       YP=USTR*W1
       W2=ALOG(YP)
       F=USTR*(A*W2+C)-UV
       FP=A*W2+C+A
       USTRF=USTR-F/FP
        ABSER=ABS((USTRF-USTR)/USTRF)
        USTR=USTRF
        IF(ABSER.LT.0.005) RETURN
  100 CONTINUE
      WRITE(6,10) YP, USTR,ABSER
   10 FORMAT(2X,'NO CONVERGENCE IN USTRF. YP,USTR,ABSER:',3F9.5)
      STOP
      END

C  *************** EDDYS ****************************

      SUBROUTINE EDDYS
C  --------------------------------------------
C    EVALUATE EDDY VISCOSITY
C
C    CALLS     : NONE
C
C    CALLED BY : KNTCS
C  ------------------------------------------

      PARAMETER (IDIM=160,JDIM=60)
      PARAMETER (IDP1=161,JDP1=61)
      PARAMETER (KFC1=241)
      common/ikp/kps(idim),kpa(idim)
      common/tmark/is1,is2,ia1,ia2
      COMMON/CORD/XX(IDP1,JDIM),YY(IDP1,JDIM)
      COMMON/SCALE/H(IDIM,JDIM)
      COMMON/IO/LOOP,NT,NTMAX,NTOUT
      COMMON/VV/VOR(IDIM,JDIM),VOROLD(IDIM,JDIM)
      COMMON/TUR1/USTRS(IDIM),USTRA(IDIM),EDDY(IDIM,JDIM)
      COMMON/GRD/IM,IM2,KFC,N
      COMMON/COE/VSC,NPL,ICST,OMG
      COMMON/VTEXT/KINS(IDIM),KENS(IDIM),KINA(IDIM),KENA(IDIM)
      COMMON/VEL/U(IDIM,JDIM),V(IDIM,JDIM)
      COMMON/DIVD/NSING
      DIMENSION YN(JDIM)
      COMMON/TYPE/LB2
      LOGICAL LB2
      DO 99 I=1,IM
      DO 99 J=1,N-1
      EDDY(I,J)=0.
   99 CONTINUE

C-- SLAT REGION

c       go to 800
      IF(LB2) THEN
      DO 105 II=1,IM
c     if(kps(ii).eq.1) go to 105
        I=II
```

```
      FMAX=0.
      YMAX=0.
      VDMX=0.

C..CALCULATE TERMS INDEPENDENT OF Y AND INNER VISC.
C  INCLUDE CORRECTION FOR THE NORMAL DIRECTION

      DO 104 J=2,NSING-1
       DX=.5*(XX(I,J)+XX(I,J+1))-XX(I,1)
       DY=.5*(YY(I,J)+YY(I,J+1))-YY(I,1)
       YN(J)=SQRT(DX*DX+DY*DY)
  104 CONTINUE
      DO 100 J=2,NSING-1
          AVOR=ABS(VOR(I,J))
          YP=USTRS(I)*YN(J)/VSC
            T1=YN(J)
          IF(YP.LE.500.)
     %      T1=YN(J)*(1.-EXP(-YP/26.))
          EDDY(I,J)=0.16*T1*T1*AVOR
          FY=AVOR*T1
          IF(FY.GT.FMAX)  YMAX=YN(J)
          IF(FY.GT.FMAX)  FMAX=FY
          UV=0.5*(U(I,J+1)+U(I,J))
          VV=0.5*(V(I,J+1)+V(I,J))
          VSPHY=(UV*UV+VV*VV)/H(I,J)
          IF(VSPHY.GT.VDMX)  VDMX=VSPHY
  100   CONTINUE

C..OUTER EDDY VISCOSITY

c     eddymax=0.
c     do 121 j=2,nsing-1
c     if(eddymax.lt.eddy(i,j))  then
c     eddymax=eddy(i,j)
c     jt=j
c     endif
c 121 continue

        IF(FMAX.EQ.0.)  GO TO 105
        FWAKE=FMAX*YMAX
        CFWAKE=0.25*YMAX*VDMX/FMAX
        IF(CFWAKE.LT.FWAKE)  FWAKE=CFWAKE
      iced=0
        DO 103 J=2,NSING-1
          FKLEB=1./(1.+5.5*(0.3*YN(J)/YMAX)**6.)
          CEDDY=0.0168*1.1*FWAKE*FKLEB
       IF( CEDDY.LT.EDDY(I,J))  iced=1
      if(iced.eq.1)   EDDY(I,J)=CEDDY
  103   CONTINUE
  105 CONTINUE

c     do 109 i=1,im
c     if(kps(i).eq.1)  then
c     do 108 j=2,nsing-1
c     eddy(i,j)=1000.*vsc
C     if(eddy(i,j).gt.0.001)  eddy(i,j)=0.001
c 108 continue
c     endif
c 109 continue

      ENDIF
  800 continue

C-- AIRFOIL REGION

      DO 205 II=1,IM
```

```
c     if(kpa(ii).eq.1) go to 205
        I=II
        FMAX=0.
        YMAX=0.
        VDMX=0.

C..CALCULATE TERMS INDEPENDENT OF Y AND INNER VISC.
C  INCLUDE CORRECTION FOR THE NORMAL DIRECTION

      yn(n-1)=0.
        DO 204 J=N-2,NSING+1,-1
         DX=.5*(XX(I,J)+XX(I,J+1))-XX(I,N)
         DY=.5*(YY(I,J)+YY(I,J+1))-YY(I,N)
       if(xx(i,j).gt.xx(1,n)) dx=0.
         YN(J)=SQRT(DX*DX+DY*DY)
      if(yn(j).le.yn(j+1)) yn(j)=yn(j+1)
  204 CONTINUE
        DO 200 J=N-2,NSING+1,-1
          AVOR=ABS(VOR(I,J))
          YP=USTRA(I)*YN(J)/VSC
            T1=YN(J)
         IF(YP.LT.500.)
     %      T1=YN(J)*(1.-EXP(-YP/26.))
          EDDY(I,J)=0.16*T1*T1*AVOR
          FY=AVOR*T1
          IF(FY.GT.FMAX) YMAX=YN(J)
          IF(FY.GT.FMAX) FMAX=FY
          UV=0.5*(U(I,J+1)+U(I,J))
          VV=0.5*(V(I,J+1)+V(I,J))
          VSPHY=(UV*UV+VV*VV)/H(I,J)
          IF(VSPHY.GT.VDMX) VDMX=VSPHY
  200   CONTINUE

C..OUTER EDDY VISCOSITY

c     eddymax=0.
c     do 221 j=n-2,nsing+1,-1
c     if(eddymax.lt.eddy(i,j)) then
c     eddymax=eddy(i,j)
c     jt=j
c      endif
c 221 continue

        IF(FMAX.EQ.0.) GO TO 205
        FWAKE=FMAX*YMAX
        CFWAKE=0.25*YMAX*VDMX/FMAX
        IF(CFWAKE.LT.FWAKE) FWAKE=CFWAKE
      iced=0
        DO 203 J=N-2,NSING+1,-1
          FKLEB=1./(1.+5.5*(0.3*YN(J)/YMAX)**6.)
          CEDDY=0.0168*1.1*FWAKE*FKLEB
      IF(CEDDY.LT.EDDY(I,J)) iced=1
       if(iced.eq.1) EDDY(I,J)=CEDDY
  203   CONTINUE
  205 CONTINUE

c     do 209 i=1,im
c     if(kpa(i).eq.1) then
c     do 208 j=n-2,nsing+1,-1
c     eddy(i,j)=1000.*vsc
C     if(eddy(i,j).gt.0.001) eddy(i,j)=0.001
c 208 continue
c     endif
c 209 continue

      DO 301 I=1,IM
```

```
      eddy(i,1)=eddy(i,2)*.15
      eddy(i,n-1)=eddy(i,n-2)*.15
c       eddy(i,1)=0.
c       eddy(i,n-1)=0.
      EDDY(I,NSING)=.5*(EDDY(I,NSING-1)+EDDY(I,NSING+1))
  301 CONTINUE
      do 302 i=1,im
      do 302 j=1,n-1
      if(eddy(i,j).ge.100.*vsc) eddy(i,j)=100.*vsc
  302 continue
       RETURN
       END

C ***************** WSURF2 ******************

      SUBROUTINE WSURF2
      PARAMETER (IDIM=160,JDIM=60)
      PARAMETER (IDP1=161)
      PARAMETER (KFC1=241)
      common/ikp/kps(idim),kpa(idim)
      COMMON/CORD/XX(IDP1,JDIM),YY(IDP1,JDIM)
      COMMON/SCALE/H(IDIM,JDIM)
      COMMON/VV/VOR(IDIM,JDIM),VOROLD(IDIM,JDIM)
      COMMON/TUR1/USTRS(IDIM),USTRA(IDIM),EDDY(IDIM,JDIM)
      COMMON/GRD/IM,IM2,KFC,N
      COMMON/COE/VSC,NPL,ICST,OMG
      COMMON/TYPE/LB2
      LOGICAL LB2

C -- SLAT REGION

      OMG2=2.*OMG
      IF(LB2) THEN
      DO 1 I=1,IM
      USTAR=USTRS(I)
      IF(VOR(I,1).LT.O.) USTAR=-USTRS(I)
      DX=.5*(XX(I,2)+XX(I,3))-XX(I,1)
      DY=.5*(YY(I,2)+YY(I,3))-YY(I,1)
      Y=SQRT(DX*DX+DY*DY)
      YP2=Y*USTRS(I)/VSC
      if(yp2.le.5.) vor(i,2)=vor(i,1)
      IF(YP2.GT.5.AND.YP2.LE.30.) VOR(I,2)=5.*USTAR/Y+OMG2
      IF(YP2.GT.30.) VOR(I,2)=2.5*USTAR/Y+OMG2
    1 CONTINUE
      ENDIF
  800 continue

C -- AIRFOIL REGION
c       go to 700

      DO 2 I=1,IM
      USTAR=USTRA(I)
      IF(VOR(I,N-1).LT.O.) USTAR=-USTRA(I)
      dx=.5*(xx(i,n-2)+xx(i,n-1))-xx(i,n)
      dy=.5*(yy(i,n-2)+yy(i,n-1))-yy(i,n)
      y=sqrt(dx*dx+dy*dy)
      yp2=y*ustra(i)/vsc
      if(yp2.le.5.) vor(i,n-2)=vor(i,n-1)
      if(yp2.gt.5.and.yp2.le.30.) vor(i,n-2)=5.*ustar/y+omg2
      if(yp2.gt.30.) vor(i,n-2)=2.5*ustar/y+omg2
    2 CONTINUE
  700 continue
      RETURN
      END
```

```
      PROGRAM LOADS

C   ***************************************************************
C     AUTHOR : C.M. WANG
C              GEORGIA INSTITUTE OF TECHNOLOGY
C
C     TAPE2   : OUTPUT FROM GEOM
C     TAPE5   : GENERAL INPUT
C     TAPE6   : GENERAL OUTPUT
C     TAPE7   : INPUT FROM PREVIOUS RUN
C     TAPE8   : OUTPUT FOR NEXT RUN
C
C   **************************************************************

      PARAMETER (IDIM=160,JDIM=60)
      PARAMETER (IDP1=161,IDP2=162)
      PARAMETER (KFC1=141,KFC2=142)
      PARAMETER (IDT2=484)
      dimension cpaa(idim)
      DIMENSION KINS(IDIM),KINA(IDIM)
      DIMENSION A(IDT2,IDT2),B(IDT2),IPVT(IDT2)
      dimension xta(idim),yta(idim),xba(idim),yba(idim)
      DIMENSION XT(IDIM),YT(IDIM),XB(IDIM),YB(IDIM)
      DIMENSION XSC(IDIM),YSC(IDIM),XAC(IDIM),YAC(IDIM)
      DIMENSION WB(IDIM+1)
      DIMENSION WIDN(600),CL(600),CD(600),CM(600)
      DIMENSION XS(IDP1),YS(IDP1),XA(IDP1),YA(IDP1)
      dimension xxc(idim,jdim),xxn(idim,jdim)
      COMMON/CORD/XX(IDP1,JDIM),YY(IDP1,JDIM)
      COMMON/AOF/SINALP,COSALP
      COMMON/DIVD/NSING
      COMMON/DANG/DPH(IDP1)
      COMMON/ANG/PHI(IDP1),TH(IDP1)
      COMMON/RGRD/R1(JDIM),R2(JDIM)
      COMMON/SMT/SGMA(KFC1)
      COMMON/COR/RP(JDIM,KFC2),RL(JDIM)
      COMMON/VV/VOR(IDIM,JDIM),VOROLD(IDIM,JDIM)
      COMMON/VEL/U(IDIM,JDIM),V(IDIM,JDIM)
      COMMON/GRD/IM,IM2,KFC,N
      COMMON/COE/VSC,NPL,ICST,OMG
      COMMON/CPC/CPA(IDP1),CPS(IDP1)
      dimension uu(idim,jdim),vv(idim,jdim)
      DIMENSION H(IDIM,JDIM)
      logical turb
      LOGICAL ANUMOT
      LOGICAL LB2,POT
      DATA AL/3.6/

C..READ GENERAL INPUTS AND ECHO THEM

      OPEN(UNIT=5,FILE='zonst.in',STATUS='OLD',FORM='FORMATTED')
      READ(5,*) ALPI,ICST,FR,IM,N
      READ(5,*) WMIN,DFMX,DRMX,KMAX,NCC
      READ(5,*) URBS,URBA,URBP,URR
      READ(5,*) NPL,NRS,NRA
      READ(5,*) DTI,DTINC,DTMAX,NTMAX
      READ(5,*) NTPL,NTOUT,NTLO
      READ(5,*) ALPMEAN,ALPAMP
      read(5,*) RE
      read(5,*) IS1,IS2,IA1,IA2
      READ(5,*) ANUMOT,TURB
      READ(5,*) KFC
      READ(5,*) LB2
      read(5,*) POT

C..INPUTS FROM GEOM
```

```
       IF(POT) RE=100000000.

      READ(3) H
      READ(3) XX,YY
      read(3) phi
      read(3) xxc,xxn
      read(3) xt,yt,xb,yb
      READ(3) R1,R2

      IM2=IM/2
      KFC=IM2+1
      PI=3.1415926535898
      VSC=AL/RE
      ALP=ALPI*PI/180.

      PHIMIN=PHI(1)
      II=1
      DO 99 I=2,IM
      IF(PHI(I).LT.PHIMIN) THEN
      II=I
      PHIMIN=PHI(I)
      ENDIF
   99 CONTINUE
      ISS=II-2
      IF(ISS.LT.0) ISS=ISS+IM

C   Construct DPH array

         DO 991 I=1,ISS
      IP1=I+1
      IM1=I-1
      IF(I.EQ.IM) IP1=1
      IF(I.EQ.1) IM1=IM
      DPH(I)=.5*(PHI(IP1)-PHI(IM1))
  991 CONTINUE
         DO 992 I=II+1,IM
      IP1=I+1
      IM1=I-1
      IF(I.EQ.IM) IP1=1
      IF(I.EQ.1) IM1=IM
      DPH(I)=.5*(PHI(IP1)-PHI(IM1))
  992 CONTINUE
      IIP1=II+1
      IF(II.EQ.IM) IIP1=1
      IIM1=II-1
      IF(II.EQ.1) IIM1=IM
      PHI2=PHI(IIP1)
      PHI1=PHI(IIM1)-2.*PI
      DPH(II)=.5*(PHI2-PHI1)
      PHI2=PHI(II)+2.*PI
      PHI1=PHI(ISS)
      IIM1=II-1
      IF(II.EQ.1) IIM1=IM
      DPH(IIM1)=.5*(PHI2-PHI1)

C..START INITIAL SOLUTION OR READ PREVIOUS ITERATION RESULTS

      READ(7) NT,N,KINS,KINA,T,DT,VOR,U,V,VORLS,VORLA,GAMAS
      if(icst.eq.4) then
      alp=(alpmean-alpamp*cos(fr*t))*pi/180.
      endif
      alpd=alp*180./pi
      print*,' *** NT = ',nt,'   T = ',t,'   ALP = ',alpd
      COSALP=COS(ALP)
      SINALP=SIN(ALP)
```

```
C  -- CONSTRUCT SLAT SURFACE GEOMETRY

      DO 1 I=1,IM
      XS(I)=XX(I,1)
      YS(I)=YY(I,1)
    1 CONTINUE
      XS(IM+1)=XS(1)
      YS(IM+1)=YS(1)
      DO 2 I=1,IM
      XSC(I)=.5*(XS(I)+XS(I+1))
      YSC(I)=.5*(YS(I)+YS(I+1))
    2 CONTINUE

C  -- CONSTRUCT AIRFOIL SURFACE GEOMETRY
c ----  Airfoil surface panels must be arranged in CCW direction

       XA(1)=XX(1,N)
       YA(1)=YY(1,N)
      DO 3 I=2,IM
        II=IM+2-I
      XA(I)=XX(II,N)
      YA(I)=YY(II,N)
    3 CONTINUE
      XA(IM+1)=XA(1)
      YA(IM+1)=YA(1)
      DO 4 I=1,IM
      XAC(I)=.5*(XA(I)+XA(I+1))
      YAC(I)=.5*(YA(I)+YA(I+1))
    4 CONTINUE

C .. VELOCITY TRANSFORM

      do 8 i=1,im
      uu(i,1)=0.
      vv(i,1)=0.
      uu(i,N)=0.
      vv(i,N)=0.
      cs1=cos(phi(i))
      sn1=sin(phi(i))
      do 8 j=2,n-1
      uze=v(i,j)*cs1-u(i,j)*sn1
      vze=u(i,j)*cs1+v(i,j)*sn1
      uu(i,j)=xxn(i,j)*vze+xxc(i,j)*uze
      vv(i,j)=xxc(i,j)*vze-xxn(i,j)*uze
    8 continue

C -- CONSTRUCT MATRIX A FOR TOTAL HEAD

      IF(LB2) THEN

C  -- FROM SLAT CONTRBUTION

      DO 11 I0=1,IM
      ALPH=ATAN2(YS(I0+1)-YS(I0),XS(I0)-XS(I0+1))
      THET=PI-ALPH
      SN=SIN(THET)
      CS=COS(THET)
      DX=XS(I0+1)-XS(I0)
      DY=YS(I0+1)-YS(I0)
      DS=SQRT(DX*DX+DY*DY)

C .. TO CONTROL POINTS ON SLAT

      DO 12 I=1,IM
      IF(I.EQ.I0) GO TO 12
```

```
      XOXI=XSC(IO)-XSC(I)
      YOYI=YSC(IO)-YSC(I)
      AA=YOYI*CS-XOXI*SN
      BB=XOXI*CS+YOYI*SN
      W1=AA*DS
      W2=AA*AA+(BB*BB-.25*DS*DS)
      A(I,IO)=-ATAN2(W1,W2)
   12 CONTINUE
      A(IO,IO)=PI

C   .. TO CONTROL POINTS ON AIRFOIL

      DO 13 I=1,IM
          I1=I+IM
      XOXI=XSC(IO)-XAC(I)
      YOYI=YSC(IO)-YAC(I)
      AA=YOYI*CS-XOXI*SN
      BB=XOXI*CS+YOYI*SN
      W1=AA*DS
      W2=AA*AA+(BB*BB-.25*DS*DS)
      A(I1,IO)=-ATAN2(W1,W2)
   13 CONTINUE
   11 CONTINUE
       ENDIF

C .. CONTRIBUTION FROM AIRFOIL PANELS

      DO 21 IO=1,IM
      IO1=IO+IM
      ALPH=ATAN2(YA(IO+1)-YA(IO),XA(IO)-XA(IO+1))
      THET=PI-ALPH
      SN=SIN(THET)
      CS=COS(THET)
      DX=XA(IO+1)-XA(IO)
      DY=YA(IO+1)-YA(IO)
      DS=SQRT(DX*DX+DY*DY)

      IF(LB2) THEN

C .. TO CONTROL POINTS ON SLAT

      DO 22 I=1,IM
      XOXI=XAC(IO)-XSC(I)
      YOYI=YAC(IO)-YSC(I)
      AA=YOYI*CS-XOXI*SN
      BB=XOXI*CS+YOYI*SN
      W1=AA*DS
      W2=AA*AA+(BB*BB-.25*DS*DS)
      A(I,IO1)=-ATAN2(W1,W2)
   22 CONTINUE
      ENDIF

C   .. TO CONTROL POINTS ON AIRFOIL

      DO 23 I=1,IM
      I1=I+IM
      IF(I1.EQ.IO1) GO TO 23
      XOXI=XAC(IO)-XAC(I)
      YOYI=YAC(IO)-YAC(I)
      AA=YOYI*CS-XOXI*SN
      BB=XOXI*CS+YOYI*SN
      W1=AA*DS
      W2=AA*AA+(BB*BB-.25*DS*DS)
      A(I1,IO1)=-ATAN2(W1,W2)
   23 CONTINUE
      A(IO1,IO1)=PI
```

```
   21 CONTINUE
      IF(.NOT.LB2) THEN
      do 25 i=1,im
       do 25 io=1,im
       a(i,io)=a(i+im,io+im)
   25 continue
      ENDIF

C -- CONSTRUCT NONHOMOGENOUS TERM: RIGHT HAND TERM

      DO 100 I=1,IM
      B(I)=2.*PI
      B(I+IM)=2.*PI
  100 CONTINUE

C.. VORTEX PANELS CONTRIBUTION

      DO 101 IO=1,IM
      DO 101 JO=1,N-1
      IOP1=IO+1
      IF(IO.EQ.IM) IOP1=1
      X2=.5*(XX(IOP1,JO+1)+XX(IOP1,JO))
      Y2=.5*(YY(IOP1,JO+1)+YY(IOP1,JO))
      X1=.5*(XX(IO,JO+1)+XX(IO,JO))
      Y1=.5*(YY(IO,JO+1)+YY(IO,JO))
      XO=.5*(X2+X1)
      YO=.5*(Y2+Y1)
      DXN=.5*(XX(IO,JO+1)-XX(IO,JO)
     %       +XX(IOP1,JO+1)-XX(IOP1,JO))
      DYN=.5*(YY(IO,JO+1)-YY(IO,JO)
     %       +YY(IOP1,JO+1)-YY(IOP1,JO))
      DN=SQRT(DXN*DXN+DYN*DYN)
      GAMA1=VOR(IO,JO)*DN
      GAMA2=VOR(IOP1,JO)*DN
      ALPH=ATAN2(Y2-Y1,X1-X2)
      THET=PI-ALPH
      SN=SIN(THET)
      CS=COS(THET)
      DX=X2-X1
      DY=Y2-Y1
      DS=SQRT(DX*DX+DY*DY)
      SS=SQRT(XO*XO+YO*YO)
      IF(SS.LE.10.) THEN
      U1=.5*(UU(IO,JO+1)+UU(IO,JO))
      U2=.5*(UU(IOP1,JO+1)+UU(IOP1,JO))
      V1=.5*(VV(IO,JO+1)+VV(IO,JO))
      V2=.5*(VV(IOP1,JO+1)+VV(IOP1,JO))
      ELSE
      U2=COSALP
      V2=SINALP
      U1=COSALP
      V1=SINALP
      ENDIF
      UP2=-U2*CS-V2*SN
      VP2=U2*SN-V2*CS
      UP1=-U1*CS-V1*SN
      VP1=U1*SN-V1*CS
      GU2=GAMA2*up2
      GV2=GAMA2*vp2
      GU1=GAMA1*Up1
      GV1=GAMA1*VP1
      FA=(GU1-GU2)/DS
      FC=(GV1-GV2)/DS

      IF(LB2) THEN
```

```
C.. TO CONTROL POINTS ON SLAT

      DO 201 I=1,IM
      XOXI=XO-XSC(I)
      YOYI=YO-YSC(I)
      AA=YOYI*CS-XOXI*SN
      BB=XOXI*CS+YOYI*SN
      FB=.5*(GU1+GU2)+BB*(GU1-GU2)/DS
      FD=.5*(GV1+GV2)+BB*(GV1-GV2)/DS
      W1=AA*DS
      W2=AA*AA+(BB*BB-.25*DS*DS)
      F1=ATAN2(W1,W2)
      W1=AA*AA+(BB-.5*DS)**2
      W2=AA*AA+(BB+.5*DS)**2
      F2=.5*ALOG(W1/W2)
      DB=FA*AA*F2+FB*F1+FC*(DS-AA*F1)+FD*F2
      B(I)=B(I)-2.*DB
  201 CONTINUE
      ENDIF

C.. TO CONTROL POINTS ON AIRFOIL

      DO 202 I=1,IM
      I1=I+IM
      XOXI=XO-XAC(I)
      YOYI=YO-YAC(I)
      AA=YOYI*CS-XOXI*SN
      BB=XOXI*CS+YOYI*SN
      FB=.5*(GU1+GU2)+BB*(GU1-GU2)/DS
      FD=.5*(GV1+GV2)+BB*(GV1-GV2)/DS
      W1=AA*DS
      W2=AA*AA+(BB*BB-.25*DS*DS)
      F1=ATAN2(W1,W2)
      W1=AA*AA+(BB-.5*DS)**2
      W2=AA*AA+(BB+.5*DS)**2
      F2=.5*ALOG(W1/W2)
      DB=FA*AA*F2+FB*F1+FC*(DS-AA*F1)+FD*F2
      B(I1)=B(I1)-2.*DB
  202 CONTINUE
  101 CONTINUE

C.. VISCOUS CONTRIBUTION

      if(.not.turb) then
      IF(LB2) THEN
      JO=1
      DO 501 IO=1,IM
      IOP1=IO+1
      IF(IO.EQ.IM) IOP1=1
      X2=.5*(XX(IOP1,JO+1)+XX(IOP1,JO))
      Y2=.5*(YY(IOP1,JO+1)+YY(IOP1,JO))
      X1=.5*(XX(IO,JO+1)+XX(IO,JO))
      Y1=.5*(YY(IO,JO+1)+YY(IO,JO))
      XO=.5*(X2+X1)
      YO=.5*(Y2+Y1)
      VORB=.5*(VOR(IOP1,jo)+VOR(IO,jo))
      ALPH=ATAN2(Y2-Y1,X1-X2)
      THET=PI-ALPH
      SN=SIN(THET)
      CS=COS(THET)
      DX=X2-X1
      DY=Y2-Y1
      DS=SQRT(DX*DX+DY*DY)

C.. TO CONTROL POINTS ON SLAT
```

```
      DO 601 I=1,IM
      XOXI=XO-XSC(I)
      YOYI=YO-YSC(I)
      AA=YOYI*CS-XOXI*SN
      BB=XOXI*CS+YOYI*SN
      W1=AA*AA+(BB-.5*DS)**2
      W2=AA*AA+(BB+.5*DS)**2
      F2=.5*ALOG(W1/W2)
      DB=VORB*F2
      B(I)=B(I)-2.*DB*AL/RE
  601 CONTINUE

C.. TO CONTROL POINTS ON AIRFOIL

      DO 602 I=1,IM
      I1=I+IM
      XOXI=XO-XAC(I)
      YOYI=YO-YAC(I)
      AA=YOYI*CS-XOXI*SN
      BB=XOXI*CS+YOYI*SN
      W1=AA*AA+(BB-.5*DS)**2
      W2=AA*AA+(BB+.5*DS)**2
      F2=.5*ALOG(W1/W2)
      DB=VORB*F2
      B(I1)=B(I1)-2.*DB*AL/RE
  602 CONTINUE
  501 CONTINUE
      ENDIF

      JO=N-1
      DO 511 IO=1,IM
      IOP1=IO+1
      IF(IO.EQ.IM) IOP1=1
      X2=.5*(XX(IOP1,JO+1)+XX(IOP1,JO))
      Y2=.5*(YY(IOP1,JO+1)+YY(IOP1,JO))
      X1=.5*(XX(IO,JO+1)+XX(IO,JO))
      Y1=.5*(YY(IO,JO+1)+YY(IO,JO))
      XO=.5*(X2+X1)
      YO=.5*(Y2+Y1)
      VORB=.5*(VOR(IOP1,jo)+VOR(IO,jo))
      ALPH=ATAN2(Y2-Y1,X1-X2)
      THET=PI-ALPH
      SN=SIN(THET)
      CS=COS(THET)
      DX=X2-X1
      DY=Y2-Y1
      DS=SQRT(DX*DX+DY*DY)
      IF(LB2) THEN

C.. TO CONTROL POINTS ON SLAT

      DO 611 I=1,IM
      XOXI=XO-XSC(I)
      YOYI=YO-YSC(I)
      AA=YOYI*CS-XOXI*SN
      BB=XOXI*CS+YOYI*SN
      W1=AA*AA+(BB-.5*DS)**2
      W2=AA*AA+(BB+.5*DS)**2
      F2=-.5*ALOG(W1/W2)
      DB=VORB*F2
      B(I)=B(I)-2.*DB*AL/RE
  611 CONTINUE
      ENDIF

C.. TO CONTROL POINTS ON AIRFOIL
```

```
      DO 612 I=1,IM
      I1=I+IM
      XOXI=XO-XAC(I)
      YOYI=YO-YAC(I)
      AA=YOYI*CS-XOXI*SN
      BB=XOXI*CS+YOYI*SN
      W1=AA*AA+(BB-.5*DS)**2
      W2=AA*AA+(BB+.5*DS)**2
      F2=-.5*ALOG(W1/W2)
      DB=VORB*F2
      B(I1)=B(I1)-2.*DB*AL/RE
  612 CONTINUE
  511 CONTINUE
      endif

 599  continue

      IF(.NOT.LB2) THEN
       do 513 i=1,im
       b(i)=b(i+im)
  513  continue
      ENDIF
      IMT2=IM*2
      IF(.NOT.LB2) IMT2=IM
      CALL SGEFA(A,IDT2,IMT2,IPVT,INFO)
      CALL SGESL(A,IDT2,IMT2,IPVT,B,0)
      IF(LB2) THEN
      DO 301 I=1,IM
      CPS(I)=B(I)
      CPAA(I)=B(I+IM)
  301 CONTINUE
      ELSE
       do 302 i=1,im
       cpaa(i)=b(i)
  302  continue
      ENDIF

c      print*,' cpaa = '
c      write(6,333) (cpaa(i),i=1,im)
c      print*,' cps = '
c      write(6,333) (cps(i),i=1,im)

  333 format(10f8.4)
      if(pot) then
      dn=(r1(n)-r1(n-1))*sqrt(h(1,n-1))
      cpaa(1)=1.-vor(1,n-1)**2*dn**2
      dn=(r1(2)-r1(1))*sqrt(h(1,1))
      cps(1)=1.-vor(1,1)**2*dn**2
      do 351 i=2,im
      ii=im+2-i
      dn=(r1(n)-r1(n-1))*sqrt(h(ii,n-1))
      cpva=1.-vor(ii,n-1)**2*dn**2
c     cpva=1.-(uu(ii,n-1)**2+vv(ii,n-1)**2)
      dn=(r1(2)-r1(1))*sqrt(h(i,1))
      cpvs=1.-vor(i,1)**2*dn**2
c      cpvs=1.-(uu(i,2)**2+vv(i,2)**2)
      print*,i,cpaa(i),cpva,cps(i),cpvs
      cps(i)=cpvs
      cpaa(i)=cpva
  351 continue
      endif

C -- CALCLUATE LOADS

      NLOAD=1
      CNP=0.
```

```
      CNF=0.
      CTP=0.
      CTF=0.
      CMP=0.
      do 180 i=2,im
      ii=im+2-i
      cpa(ii)=.5*(cpaa(i)+cpaa(i-1))
  180 continue
      cpa(1)=.5*(cpaa(1)+cpaa(im))
      write(4,*) t,nt,alpd,IM,Ib2
      write(4,1225) (cpa(i),i=1,im)
      write(4,1225) (cps(i),i=1,im)
 1225 format(10e12.5)

      CMF=0.
      DO 190 I=1,IM
      WB(I)=VOR(I,N-1)-2.*OMG
  190 CONTINUE
      WB(IM+1)=WB(1)
      DO 191 I=1,IM
      dx=xa(i+1)-xa(i)
      dy=ya(i+1)-ya(i)
      dcnp=cpaa(i)*dx
      dctp=-cpaa(i)*dy
      dcmp=dcnp*xac(i)-dctp*yac(i)
      cnp=cnp+dcnp
      ctp=ctp+dctp
      cmp=cmp+dcmp

      go to 191

      if(.not.turb) then
       wba=.5*(wb(i)+wb(i+1))
       dcnf=wba*dy
       dctf=wba*dx
       dcmf=dcnf*xac(i)-dctf*yac(i)
       cnf=cnf+dcnf
       ctf=ctf+dctf
       cmf=cmf+dcmf
      endif
  191 CONTINUE

c     print*,(dph(i),i=1,im)
      al=3.6
      CN=CNP/AL+2./RE*CNF
      CT=CTP/AL+2./RE*CTF
       CL(NLOAD)=CN*COSALP-CT*SINALP
       CD(NLOAD)=CN*SINALP+CT*COSALP
       CM(NLOAD)=CMP/(AL*AL)+2./(RE*AL)*CMF
c      PRINT*, '  NO VISCOUS FORCES '
      print*,'  '
      print*,' *** Airfoil ***'
      PRINT*,' CL CD CM = ',CL(NLOAD),CD(NLOAD),
     %       CM(NLOAD)
      write(32,*) nt,t,cl(nload),cD(Nload),-Cm(Nload),alpD
      cnp=0.
       cnf=0.
       ctp=0.
       ctf=0.
       cmf=0.
        cmp=0.
      do 186 i=1,im
       wb(i)=vor(i,1)-2.*omg
  186 continue
       wb(im+1)=wb(1)
       dxxx=0.
```

```
      dyyy=0.
      do 196 i=1,im
      dx=xs(i+1)-xs(i)
      dy=ys(i+1)-ys(i)
      dcnp=cps(i)*dx
      dctp=-cps(i)*dy
      dcmp=dcnp*xsc(i)-dctp*ysc(i)
      cnp=cnp+dcnp
      ctp=ctp+dctp
      cmp=cmp+dcmp

      go to 196

      if(.not.turb) then
       wba=.5*(wb(i)+wb(i+1))
       dcnf=wba*dy
       dctf=wba*dx
      dcmf=dcnf*xsc(i)-dctf*ysc(i)
       cnf=cnf+dcnf
       ctf=ctf+dctf
      cmf=cmf+dcmf
      endif
196   continue
      cn=cnp/al+2./re*cnf
      ct=ctp/al+2./re*ctf
      cl(1)=cn*cosalp-ct*sinalp
      cd(1)=cn*sinalp+ct*cosalp
      cm(1)=cmp/(al*al)+2./(re*al)*cmf
      print*,'  '
      print*,' *** Slat *** '
      print*,' Cl CD CM = ',cl(1),cd(1),cm(1)
      write(33,*) nt,t,cl(1),cD(1),-CM(1),alpD
      STOP
      END
```